

UNCLASSIFIED

AD NUMBER

AD476192

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited. Document partially illegible.

FROM:

Distribution authorized to U.S. Gov't. agencies and their contractors;
Administrative/Operational Use; DEC 1965. Other requests shall be referred to Army Missile Command, Redstone Arsenal, AL.

AUTHORITY

usamc ltr 23 aug 1971

THIS PAGE IS UNCLASSIFIED

ARPA ORDER NO. 347

PROJECT CODE NO. 7400

GENERAL MOTORS CORPORATION

TECHNICAL REPORT

ON

A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILBRIUM STREAMTUBE FLOW

Sponsored By

ADVANCED RESEARCH PROJECTS AGENCY

Monitored By

U.S. ARMY MISSILE COMMAND

CONTRACT NO. DA-01-021-AMC-11359(Z)

HYPERVELOCITY RANGE RESEARCH PROGRAM

A PART OF PROJECT "DEFENDER"

GM DEFENSE RESEARCH LABORATORIES

SANTA BARBARA, CALIFORNIA



AEROSPACE OPERATIONS DEPARTMENT



TR45-01P

DECEMBER 1965

DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST
QUALITY AVAILABLE.

COPY FURNISHED CONTAINED
A SIGNIFICANT NUMBER OF
PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

NOTICE OF AVAILABILITY OF SOURCE PROGRAM

The source deck for the computer program listed herein will be duplicated and transmitted upon written request. Any deck supplied in this manner will reflect the status of the codes current at the date of this report and will not incorporate any further developments. Further, since the computing methods have been checked out over only a limited set of systems and thermodynamic conditions, their functioning over a general range of inputs cannot be assured.

ARPA ORDER NO. 347

Copy No. 

PROJECT CODE NO. 7400

GENERAL MOTORS CORPORATION

TECHNICAL REPORT

ON

A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILBRIUM STREAMTUBE FLOW

Tung Chen and Alan Q. Eschenroeder

THIS RESEARCH WAS SUPPORTED BY THE
ADVANCED RESEARCH PROJECTS AGENCY
AND WAS MONITORED BY THE
U.S. ARMY MISSILE COMMAND
REDSTONE ARSENAL, ALABAMA

GM DEFENSE RESEARCH LABORATORIES

SANTA BARBARA, CALIFORNIA



AEROSPACE OPERATIONS DEPARTMENT

CONTRACT NO. DA-01-021-AMC-11359(Z)
HYPERVELOCITY RANGE RESEARCH PROGRAM
A PART OF PROJECT "DEFENDER"

TR65-01P

DDC AVAILABILITY NOTICE

Qualified requesters may obtain
copies of this report from DDC

DECEMBER 1965

FOREWORD

This report is one of a series of related papers covering various aspects of a broad program to investigate the flow-field variables associated with hypersonic-velocity projectiles in free flight under controlled environmental conditions. The experimental research is being conducted in the Flight Physics Range of GM Defense Research Laboratories, General Motors Corporation, and is supported by the Advanced Research Projects Agency under Contract No. DA-01-021-AMC-11359(Z). It is intended that this series of reports, when completed, will provide a background of knowledge of the phenomena involved in the basic study and thus aid in a better understanding of the data obtained in the investigation.

ABSTRACT

A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

CONTENTS

<u>Section</u>	<u>Page</u>
FOREWORD	ii
ABSTPACT	iii
LIST OF MATHEMATICAL SYMBOLS	vi
I INTRODUCTION	1
II GOVERNING EQUATIONS	4
III NUMERICAL PROCEDURE	10
A. Runge-Kutta-Richardson Method	13
B. Perturbation Method for Chemical Reactions near Equilibrium	20
IV PROGRAM DESCRIPTION	24
REFERENCES	27
APPENDIXES	
Appendix A: Input Formats	A-1
Appendix B: Flow Diagram	B-1
Appendix C: FORTRAN Symbols and Program Listing	C-1

LIST OF MATHEMATICAL SYMBOLS

NOTE: PRIMES DENOTE DIMENSIONAL QUANTITIES

A'	cross-sectional area of stream tube $A = A'/1.0$
A_{ij}	denotes vibrational coupling of j^{th} species to i^{th} reaction
BC	boundary condition for the streamtube
C	number of elements
C_{pj}	specific heat of j^{th} species $C_{pj} = \frac{C_{pj}'}{R_0'}$
$E_{j\ell}$	energy of ℓ^{th} electronic level of j^{th} species
E_{vj}	energy of v^{th} vibrational level of j^{th} species
ΔF_i^0	change in standard free energy for the i^{th} reaction
$g_{j\ell}$	degeneracy of ℓ^{th} electronic level of j^{th} species
H'	enthalpy of mixture; $H = \frac{H' MW_0'}{R_0' T_0'}$
h_j	enthalpy of j^{th} species including heat of formation $h_j = \frac{h_j'}{R_0' T_0'}$
h_j^0	energy of formation of j^{th} species; $h_j^0 = \frac{h_j'}{R_0' T_0'}$
k_{Fi}', k_{Bi}'	forward and backward reaction rate coefficients
K_i	equilibrium constant of i^{th} reaction
k	Boltzmann constant
L'	reference length in stream tube
M_j	denotes j^{th} species
MW'	molecular weight of gas mixture; $MW = \frac{MW'}{MW_0'}$
m_j	number of electronic levels included for j^{th} species
N_j	number of vibrational energy levels of j^{th} species
n_j	number of atoms per species

N_0	Avagadro number
p'	pressure; $p = \frac{p'}{p'_0(U'_0)^2}$
Q_{ij}	mole-volumetric rate of production of M_j from reaction i ;

$$Q_{ij} = Q'_{ij} \frac{MW'_0 L'}{U'_0 p'_0}$$

$Q(T)$	vibrational partition function
R'_0	universal gas constant ($R'_0 = 1.98647 \text{ cal/mole-}^\circ\text{K}$)
r	number of chemical reactions considered
s	number of species in the gas mixture
t	time
T' or T_T'	translational temperature; $T = \frac{T'}{T'_0}$
T_{Fj}'	parameter of j^{th} species having units of temperature; $T_{Fj} = \frac{T_{Fj}'}{T'_0}$
T_{mj}'	parameter of j^{th} species having units of temperature; $T_{mj} = \frac{T_{mj}'}{T'_0}$
U_j'	characteristic probability parameter of j^{th} species with units of temperature; $U_j = \frac{U_j'}{T'_0}$
U'	velocity; $U = \frac{U'}{U'_0}$
V_j	vibrational coupling factor of j^{th} species
y'	coordinate along streamtube: $y = \frac{y'}{L'}$
α_{jk}	atoms of k^{th} element per molecule of j^{th} species
γ	ratio of specific heats
γ_j'	concentration of j^{th} species in moles per unit mass; $\gamma_j = \gamma'_j(MW'_0)$
ϵ_j'	vibrational energy of j^{th} species; $\epsilon_j = \frac{\epsilon_j'}{R'_0 T'_0}$
θ_{rj}'	characteristic rotational temperature of j^{th} species; $\theta_{rj} = \frac{\theta_{rj}'}{T'_0}$
θ_{vj}'	characteristic vibrational temperature of j^{th} species; $\theta_{vj} = \frac{\theta_{vj}'}{T'_0}$
Λ	nondimensionalizing term,

$$\Lambda = \frac{(U'_0)^2 MW'_0}{R'_0 T'_0}$$

- λ_j vibrational relaxation distance of j^{th} species; $\lambda_j = \frac{\lambda_j'}{L'}$
- μ_j chemical potential of j^{th} species
- ν_j, ν_j^* stoichiometric coefficients of j^{th} species in i^{th} reaction
- ρ' density; $\rho = \frac{\rho'}{R'}$
- τ_j vibrational relaxation time of j^{th} species
- χ_i degree of nonequilibrium of i^{th} reaction
- ω_j vibrational frequency of j^{th} molecule

Subscripts

- ∞ refers to vibrational equilibrium
- \circ reference condition
- i pertaining to i^{th} reaction
- j pertaining to j^{th} species
- l pertaining to l^{th} electronic level
- v pertaining to v^{th} vibrational level

Primes denote dimensional quantities

NOTE: INITIAL CONDITIONS MUST OBEY EQUATION OF STATE WHICH EMPLOY THE CONSTANTS USED IN THE PROGRAMS

$$p_o' = \frac{p_o' T_o'}{MW_o'} (8.313405 \times 10^7)$$

$$p_o' \text{ dynes/cm}^2$$

$$T_o' \text{ } ^\circ K$$

$$\rho_o' \text{ gms/cm}^3$$

$$MW_o' \text{ gms/mole}$$

I. INTRODUCTION

Current studies of hypersonic aerothermodynamics require rapid solutions for high-temperature properties of gas mixtures in the chemical relaxation region of a flow field. Normal and bow shock wave computer programs have been written by the Cornell Aeronautical Laboratories^{1,2} to calculate the inviscid nonequilibrium-flow-field gas properties behind a normal shock wave expanding along a constant-area stream tube or in the shock layer of a blunt body at hypersonic speed, but frequently the nonequilibrium gas properties must be determined behind a shock wave along an expanding stream tube with given streamwise cross-section area or pressure variation. Hypersonic blunt-body flow fields and nozzle flows of reacting gas are examples of technically significant problems of this nature. To fulfill these requirements, an inviscid nonequilibrium streamtube flow computer program has been generated from CAL's normal shock wave computer program.²

The frozen normal shock calculations to get the post-shock properties in the CAL Normal Shock Wave Program has been replaced by reading in all the initial conditions at the starting point of the streamtube expansion. We allow an additional variable, the cross-sectional area of the streamtube denoted by the symbol A . This makes the gas density, ρ , in the continuity equation a function of both flow velocity, U , and cross-sectional area, A . Hence, we need one more differential equation. the

equation of boundary condition, to solve for this additional variable, A . The boundary condition for this streamtube computer program can be the streamwise variation of pressure, cross-sectional area, density or velocity.

The integration method, Runge-Kutta method, which is used in the CAL Normal Shock Program, of solving the ordinary differential equations is modified with the Richardson's extrapolation to increase the computing speed (Section III-A). With this modification the running time is reduced by more than one half. In addition, an accuracy check is included in the modified Runge-Kutta Method. Therefore, experience with specific previous problems is unnecessary for the choice of step size needed for accurate results. This saves computing time which would otherwise be required for test runs to judge the accuracy of the results.

The running time for a chemical nonequilibrium-flow-field calculation is usually quite long in a region where one or more of the reactions are near chemical equilibrium. This occurs because the concentrations of the species which are near chemical equilibrium begin oscillations around the local equilibrium concentration values of that species; hence the numerical integration step size should be very small to get accurate results. An artificial perturbation on the degree of nonequilibrium, χ_i , of each reaction is used in this program to save computing time near equilibrium. The procedure is described in Section III-B.

This streamtube computer program can be used to calculate the nonequilibrium inviscid high-temperature properties of gas mixtures for quasi-one-dimensional flows and can be run near equilibrium. The boundary conditions of this streamtube computer program are the streamwise variations of pressure, density, velocity or cross-sectional area of the flow field.

This report briefly discusses the governing equations and the numerical procedures used in this program. The general operating procedures are also presented.

II. GOVERNING EQUATIONS

In the streamtube program, the variables are velocity (U), pressure (p), density (ρ), cross-sectional area (A), species concentration (γ_j) and vibrational energy (ϵ_j). The variables U , p , ρ , and A are all in nondimensional forms;

$$U = \frac{U'}{U'_0} \quad (2-1)$$

$$p = \frac{p'}{\rho'_0 U'^2_0} \quad (2-2)$$

$$\rho = \frac{\rho'}{\rho'_0} \quad (2-3)$$

$$A = \frac{A'}{L'^2_0} \quad (2-4)$$

Primed quantities are dimensional and subscript "o" denotes reference conditions.

The conservation equations on a body-fixed coordinate system are^{1, 2}

Continuity Equation

$$\frac{d\rho}{dy} = -\frac{\rho}{U} \frac{dU}{dy} - \frac{\rho}{A} \frac{dA}{dy} \quad (2-5)$$

Momentum Equation

$$\frac{dp}{dy} = -\rho U \frac{dU}{dy} \quad (2-6)$$

Energy Equation

$$\Lambda U \frac{dU}{dy} = - \sum_{\alpha=c+1}^s \gamma_{\alpha} (n_{\alpha}-1) \frac{d\epsilon_{\alpha}}{dy} - \Lambda \frac{MW}{\rho} \left(\frac{dp}{dy} - \frac{h}{\rho} \frac{d\rho}{dy} \right) \quad (2-7)$$

$$+ \sum_{\alpha=1}^s \gamma_{\alpha} C_{p\alpha} - \sum_{\alpha=1}^s \sum_{j=1}^s \left(\frac{h_j}{s} - \Lambda \frac{MW^2}{\rho} \gamma_{\alpha} C_{p\alpha} \right) \frac{d\gamma_{\alpha}}{dy}$$

Conservation of j^{th} Species

$$\frac{d\gamma_j}{dy} = \frac{1}{\rho U} \sum_{i=1}^r Q_{ij} \quad j = c+1, \dots, s \quad (2-8)$$

Conservation of k^{th} Element

$$\sum_{j=1}^s \alpha_{jk} \frac{d\gamma_j}{dy} = 0 \quad k = 1, \dots, c \quad (2-9)$$

Boundary Condition

The equation for the streamtube boundary condition can be any one of the following:

$$\frac{dA}{dy} = \frac{dBC_1(y)}{dy} \quad (2-10)$$

$$\frac{dp}{dy} = \frac{dBC_2(y)}{dy} \quad (2-11)$$

$$\frac{d\rho}{dy} = \frac{dBC_3(y)}{dy} \quad (2-12)$$

$$\frac{dU}{dy} = \frac{dBC_4(y)}{dy} \quad (2-13)$$

Where $BC_i(y)$ represents U, p, ρ or A as a function of y either in a polynomial or tabular form.

The temperature is computed from the equation of state.

$$T = \frac{p \cdot MW}{\rho} \quad (2-14)$$

The reversible chemical reactions in the stream tube program involving the species M_j ($j = 1, 2, 3, \dots, s$) are represented by

$$\sum_{j=1}^s \nu_{ij} M_j \xrightleftharpoons[k_{B_i}']{k_{F_i}'} \sum_{j=1}^s \nu_{ij}^* M_j \quad (2-15)$$

where ν_{ij} and ν_{ij}^* are the stoichiometric coefficients.

Let

$$\beta_{ij} = \nu_{ij}^* - \nu_{ij} \quad (2-16)$$

$$\beta_i = \sum_{j=1}^s \beta_{ij} \quad (2-17)$$

and

$$\nu_i = \sum_{j=1}^s \nu_{ij} \quad (2-18)$$

The equilibrium constant K_i' can be expressed as a function of the change in free energy

$$K_i' = e^{-\frac{\Delta F_i^0}{R_o' T_o'}} (R_o' T_o')^{-\beta_i} \quad (2-19)$$

where

$$\frac{\Delta F_i^0}{R_o' T_o'} = \sum_{j=1}^s \beta_{ij} \frac{\mu_j^0}{T} \quad (2-20)$$

6 where μ_j^0 is the chemical potential of the j^{th} species at standard pressure.

Based on the molecular model of a simple harmonic oscillator, μ_j^0 can be given as

$$\frac{\mu_j^0}{T} = - \left\{ a_j + \frac{5+2(n_j-1)}{2} \ln T + (n_j-1) \ln \left[\frac{1}{1 - e^{-(\theta_j/T)}} \right] + \ln \left[\frac{\sum_{k=1}^{\infty} g_{jk} e^{-(E_{jk}/T)}}{g_{j,1}} \right] \right\} + \frac{h_j^0}{T} \quad (2-21)$$

where

$$a_j = b_j + \frac{5+2(n_j-1)}{2} \ln T_0' \quad (2-22)$$

and

$$b_j = \frac{3}{2} \ln \left(\frac{2\pi m_{jk}}{h^2} \right) + \ln k - (n_j-1) \ln \theta_j + \ln g_{j,1} \quad (2-23)$$

In the equation of conservation of j^{th} species, the molar volumetric rate of production of species j from reaction i , Q_{ij} , can be calculated² by

$$Q_{ij}' = W_I \left\{ [\beta_{ij} k_{Fi}' (\rho')^2 \prod_{j=1}^3 (\gamma_j')^{\nu_{ij}'}] \chi_i \right\} + D_I \left\{ \left[\frac{\beta_{ij} k_{Fi}' (\rho')^2}{MN'} \prod_{j=1}^3 (\gamma_j')^{\nu_{ij}'} \right] \chi_i \right\} + Z_I \left\{ [\beta_{ij} k_{Fi}' (\rho')^2 \gamma_j' \sum_{j=1}^3 (\beta_{ij} + 1) \nu_{ij} \gamma_j'] \chi_i \right\} \quad (2-24)$$

where W_I , D_I , and Z_I may be 1 or 0 and χ_i indicates the degree of nonequilibrium. The first term in Eq. (2-24) ($W_I=1, D_I=Z_I=0$) provides for a particular nonreactive collider in the chemical reactions, such as



The second term alone in Eq. (2-24) ($D_I=1, W_I=Z_I=0$) combines all collision partners, such as $\text{O}_2 + M \rightleftharpoons 2\text{O} + M$. The third term alone in Eq. (2-24)

($Z_I=1, W_I=D_I=0$) is included if some subgroup of species are lumped together as a single collision partner, such as $\text{NO} + M \rightleftharpoons \text{N} + \text{O} + M$.

The variation of vibrational energy can be expressed as follows:^{1,2}

A. Nonpreferential Model

Vibrational Energy Equation

$$\frac{d\epsilon_j}{dy} = \frac{\epsilon_{j\infty} - \epsilon_j}{\lambda} + \left\{ \left[\frac{\theta_{vj}}{e^{\theta_{vj}/T_{vj}} - 1} - \frac{N_j \theta_{vj}}{e^{N_j \theta_{vj}/T_{vj}} - 1} \right] - \epsilon_j \right\} \times$$

$$\sum_{i=1}^f \frac{A_{ij}}{\gamma_j \rho U} \frac{Q_{ij}}{\chi_i} - \left\{ \left[\frac{1}{2} (N_j - 1) \theta_{vj} \right] - \epsilon_j \right\} \sum_{i=1}^f \frac{A_{ij} \theta_{ij}}{\gamma_j \rho U} \frac{1 - \chi_i}{\chi_i}$$

$$j = f+1, \dots, g \quad (2-25)$$

Vibrational Coupling Factor

$$V_j = \frac{1}{N_j} \frac{1 - e^{-N_j \theta_{vj}/T_{vj}}}{e^{\theta_{vj}/T_{vj}} - 1} \frac{e^{\theta_{vj}/T_{vj}} - 1}{e^{\theta_{vj}/T} - 1} \quad (2-26)$$

B. Preferential Model

Vibrational Energy Equation

$$\frac{d\epsilon_j}{dy} = \frac{\epsilon_{j\infty} - \epsilon_j}{\lambda_j} + \{ \bar{E}_j - \epsilon_j \} \sum_{i=1}^f \frac{A_{ij}}{\gamma_j \rho U} \frac{Q_{ij}}{\chi_i} - \{ \bar{G}_j - \epsilon_j \} \times$$

$$\sum_{i=1}^f \frac{A_{ij} Q_{ij}}{\gamma_j \rho U} \frac{1 - \chi_i}{\chi_i} \quad (2-27)$$

where

$$\bar{E}_j = \frac{1}{Q(T_{Fj})} \sum_v^{N_j} E_{vj} e^{-\frac{E_{vj}}{RT_{Fj}}} \quad (2-28)$$

$$\bar{G}_j = \frac{1}{Q(-U_j)} \sum_v^{N_j} E_{vj} e^{-\frac{E_{vj}}{RU_j}} \quad (2-29)$$

and Q is the vibrational partition function.

Vibrational Coupling Factor

$$V_j = \frac{Q(T)Q(T_{Fj})}{Q(T_{vj})Q(-U_j)} \quad (2-30)$$

The enthalpy and specific heat are given as

$$h_j = \left[\frac{5+2(n_j-1)}{2} \right] T + \frac{(n_j-1)\theta_{vj}}{e^{\theta_{vj}/T} - 1} + \frac{\sum_{i=1}^m \epsilon_{ji} g_{ji} e^{-(\epsilon_{ji}/T)}}{\sum_{i=1}^m g_{ji} e^{-(\epsilon_{ji}/T)}} + h_j^0$$

$$j = 1, 2, \dots, f, \quad g+1, \dots, s \quad (2-31)$$

where n_j is 1 or 2 and denotes the number of atoms per molecules.

$$C_{pj} = \left[\frac{5+2(n_j-1)}{2} \right] + (n_j-1) \left(\frac{\theta_{vj}}{T} \right)^2 \frac{e^{\theta_{vj}/T}}{(e^{\theta_{vj}/T} - 1)^2} +$$

$$\frac{\left[\sum_{i=1}^m g_{ji} e^{-\frac{\epsilon_{ji}}{T}} \right] \left[\sum_{i=1}^m g_{ji} \left(\frac{\epsilon_{ji}}{T} \right)^2 e^{-\frac{\epsilon_{ji}}{T}} \right] - \left[\sum_{i=1}^m \epsilon_{ji} g_{ji} e^{-\frac{\epsilon_{ji}}{T}} \right] \left[\sum_{i=1}^m \frac{g_{ji} \epsilon_{ji}}{T^2} e^{-\frac{\epsilon_{ji}}{T}} \right]}{\left[\sum_{i=1}^m g_{ji} e^{-(\epsilon_{ji}/T)} \right]^2} \quad (2-32)$$

For vibration equilibrium the second term in Eq. (2-31) is written as

$$(n_j-1) \frac{\theta_{vj}}{e^{(\theta_{vj}/T_{vj})} - 1}$$

and the second term in Eq. (2-32) should be omitted.

III NUMERICAL PROCEDURE

The procedure for solving a set of simultaneous ordinary differential equations of first order is to solve for the derivatives of every variable with respect to y first and then integrate streamwise to get the properties at every point. The modified method of elimination (Guass' method) is used to solve the set of simultaneous linear equations for the derivatives of every unknown with respect to y . Then a fourth order Runge-Kutta method with Richardson's modification (See Section III-A) is used to integrate along the streamtube. This integration method, Runge-Kutta-Richardson's Method, has the following two advantages:

1. It is very easy to put in a mathematical test for an accuracy check without spending excessive computing time. With this check to determine the starting integration step size, uncertainties in the choice of a proper initial integration step size are eliminated. This saves considerable computing time.
2. With the residue modification, the integration step size can be much larger to get results of a given accuracy. This will usually increase computing speed by a factor of 3 or 4.

In the relaxation zone of a complex reacting flow field such as an air flow, some of the chemical reactions approach local quasi-equilibria (i.e., $X_i \rightarrow 0$). In this region, the concentration of some or all species may fluctuate around the quasi-equilibrium concentration of that species.

Hence, the integration step must be restricted to a very small size to guarantee the accuracy. This slows down the computation very much in the local quasi-equilibrium region. A method involving artificial perturbations on the chemical reaction near equilibrium (See Section III-B) has been developed. With this method the nonequilibrium program is used for rapid calculation through the local quasi-equilibrium region. In the perturbation method, the outer bandwidth around zero of ± 0.1 and the inner bandwidth around zero of ± 0.05 are recommended. Section III-B gives definitions of the bandwidths in the description of the method.

The boundary condition for this streamtube program is either in polynomial or in tabular form. When it is in a tabular form, the derivative of the boundary condition function is first found with respect to y by a divided differences method. Then it is integrated to get the boundary condition function for each y . To solve the set of simultaneous differential equations, Eq. (2-5) to Eq. (2-9), the derivative boundary condition function with respect to y must be evaluated. And also, the derivatives of every variable are solved with respect to y first. Integration is then carried out to get the variable values for each y . That is why the derivative of the boundary condition function is obtained from the table instead of interpolating directly from the table for each interval. In this way, the boundary values obtained by differentiation and then integration will usually not be exactly the same as the boundary condition values directly interpolated from the table. For some cases, a boundary condition function, such as

pressure, may decrease with y in a few orders of magnitude from the beginning to the end of a region we considered. In this case, the cumulative errors will be very large near the end point, if there are very small errors in the beginning due to the indirect way to get the boundary condition value. To correct this kind of error, the boundary condition function is also interpolated from the table directly at every point and is used as the current value instead of the value from integration after each interval. With this correction, the boundary condition will follow the prescribed boundary condition function.

A. RUNGE-KUTTA-RICHARDSON METHOD

1. Introduction

The Runge-Kutta method is one of the popular methods for solving first-order ordinary differential equations numerically.^{3,4} For higher-order differential equations, every higher-order differential equation is usually represented by a set of first-order differential equations. Thus, the Runge-Kutta technique is also applicable to higher-order ordinary differential equations.

When the Runge-Kutta method is employed to solve differential equations, the accuracy of the results will vary inversely with the integration step size. To ascertain the use of the proper integration step size, after each step is completed the step size is halved and the integration over the whole interval just completed is repeated. Then, by comparing the result of the full-integration step size with the one with half-step size and integrated twice, we can find the deviation of the result by Runge-Kutta Method from the exact value for this step and see if the step size we just used is proper or not in this region. However, this kind of mathematical check of the Runge-Kutta method usually triples the computing time.

In aerospace fields, many problems require the numerical solution of a set of differential equations; for example, the calculation of the gas properties for a nonequilibrium flow field around a blunt body. For this problem, CAL has a computer program.² In that program, they use the

Runge-Kutta method to solve a set of ordinary differential equations numerically. Since this is a very complicated problem and requires very much computing time, they omit the mathematical check just mentioned and choose the integration step size by experience to save the computing time. When this computer program is applied to an unfamiliar problem with new initial conditions, the integration step size must be chosen by iteration. This iteration procedure consumes much computing time and turnaround time.

In this section a method is derived for modifying the Runge-Kutta results by Richardson's extrapolation. Furthermore, the mathematical check is added to optimize the integration step size. Since the proper integration step size with the modification is larger than the one without modification, the total computing time is of the same order or even shorter than the unmodified one without mathematical check. Thus, when this modification is used, we can get more accurate results with less computing time.

2. Richardson's Extrapolation

The Runge-Kutta method is based on the Taylor series expansion.

We have a differential equation

$$\frac{dy}{dx} = f(x, y) \quad (3-1)$$

with the initial condition $y = y_0$ at $x = x_0$

Suppose the solution of Eq. (3-1) is

$$y = F(x) \quad (3-2)$$

We can use the Taylor series to expand Eq. (3-2) as a function of F and x to the N^{th} order.

$$y = F(x_0) + h F'(x_0) + \cdots + \frac{h^N}{N!} F^{(N)}(x_0) + R \quad (3-3)$$

where

x_0 = the initial value of x

$h = x - x_0$

$$R = \frac{h^{N+1}}{(N+1)!} F^{(N+1)}(\xi)$$

ξ = a point between x and x_0 .

When we use the fourth order Runge-Kutta method, $N = 4$, the remainder, R , we neglected is

$$R = \frac{h^5}{5!} F^{(5)}(\xi) \quad (3-4)$$

The deviation from the exact value for each step is

$$(\text{Dev})_1 = \frac{F^{(5)}(\xi) \times h^5}{5!} \quad (3-5)$$

If the step size, h , we used is not very large we can consider this deviation to be proportional to the fifth power of the step size, h , only.

Or we can write

$$(\text{Dev})_1 = K \times h^5 \quad (3-6)$$

where K is a constant.

If we choose half the step size and integrate it twice

$$(\text{Dev})_2 = K \left(\frac{h}{2}\right)^5 \times 2 = K \times \frac{h^5}{16} \quad (3-7)$$

The residue, the difference between the deviation from full integration step size, $(Dev)_1$, to the one with half integration step size, $(Dev)_2$, is

$$\begin{aligned}
 \text{Residue} &= (Dev)_1 - (Dev)_2 \\
 &= K \left(h^5 - \frac{h^5}{16} \right) \\
 &= \frac{15}{16} K h^5 \\
 &= 15 (Dev)_2
 \end{aligned}
 \tag{3-8}$$

Hence subtracting one fifteenth the residue from the result obtained by using half integration step size and integrating twice will recover the exact value, if the deviation for each step is proportional to the fifth power of the step size. This modification is the Richardson's extrapolation.⁴

The procedure of using Richardson's extrapolation is as follows:

Suppose we know the function, $f(x, y)$, in Eq. (3-1). Using the Runge-Kutta method, we integrate Eq. (3-1) from x_0 to $x_0 + h$ with a full integration step size, h . We get a result, $y = y_0 + y_1$ at $x = x_0 + h$. If we integrate Eq. (3-1) again from x_0 to $x_0 + h$ with a half integration step size, $h/2$. We may get another result, $y = y_0 + y_2$ at $x = x_0 + h$. If the step size, h , is small enough, y_1 may be equal to y_2 . But, usually they are not equal.

From the definition of residue, Eq. (3-8), we have

$$\text{Residue} = y_1 - y_2 \quad (3-9)$$

Hence, with the Richardson's extrapolation, the result should be

$$\begin{aligned} y &= y_0 + y_2 - (\text{Dev})_2 \\ &= y_0 + y_2 - \text{Residue}/15 \\ &= y_0 + y_2 - (y_1 - y_2) / 15 \end{aligned} \quad (3-10)$$

This result should be very close to the exact solution of Eq. (3-1).

When we use the Runge-Kutta-Richardson method to integrate a differential equation, if we can make $\text{Residue}/y_1 = (y_1 - y_2)/y_1 < \epsilon$, then the deviation of the result from the exact solution should be much smaller than ϵ . Hence, we can use the residue test, $\text{Residue}/y_1 < \epsilon$, as a procedure to control the integration step size. The residue test takes very little computing time.

3. Example

Suppose we have a differential equation

$$\frac{dy}{dx} = 1 - 2x + 3x^2 - 4x^3 + 5x^4 - 6x^5 \quad (3-11)$$

with the initial condition

$$y = 1, \text{ at } x = 0$$

The solution of Eq. (3-11) is

$$y = 1 + x - x^2 + x^3 - x^4 + x^5 - x^6 \quad (3-12)$$

We can use the Runge-Kutta method and also the one with the Richardson's extrapolation to solve Eq. (3-11) numerically at a different step size to see how much the Richardson's extrapolation will improve the results.

In Table 1, $\Delta X = 0.8$, the results obtained by strictly Runge-Kutta method are accurate to the 3rd digit. The results with the Richardsons' extrapolation are accurate up to 6 or 7 digits.

In Table 2, $\Delta X = 3.2$, the results obtained by the Runge-Kutta method with Richardson's extrapolation are even better than the results by strictly Runge-Kutta method with $\Delta X = 0.8$.

4. Conclusion

From the above example, we can see that in Table 2, the ΔX we used is four times the ΔX we used in Table 1. That means the Runge-Kutta method with the Richardsons' extrapolation we used in Table 2 consumes the same order of computing time as the results we obtained from the unmodified Runge-Kutta method in Table 1. Hence, with the Richardsons' extrapolation we improve the results and impose a mathematical check to assure accuracy with the same order of computing time.

The space needed for this modification with the residue test is very small in an actual computer program.

TABLE 1. NUMERICAL SOLUTION OF EQ. (3-11) (DX=0.8)

X	Y(EXACT)	Y(R-K)	Y(R-K-R)
0.0000000E 00	1.0000000E 00	1.0000000E 00	1.0000000E 00
8.0000000E-01	1.3279360E 00	1.3088213E 00	1.3279360E 01
1.6000000E 00	-8.7090555E 00	-8.6128207E 01	-8.7090547E 01
2.4000000E 00	-1.3319032E 02	-1.3344428E 02	-1.3319032E 02
3.2000000E 00	-8.1632700E 02	-8.1679668E 02	-8.1632698E 02
4.0000000E 00	-3.2749997E 03	-3.2757505E 03	-3.2749995E 03
4.8000000E 00	-1.0120038E 04	-1.0121137E 04	-1.0120039E 04
5.6000000E 00	-2.6166250E 04	-2.6167761E 04	-2.6166251E 04
6.4000000E 00	-5.9431183E 04	-5.9433173E 04	-5.9431184E 04
7.2000000E 00	-1.2232264E 05	-1.2232518E 05	-1.2232264E 05
8.0000000E 00	-2.3301495E 05	-2.3301809E 05	-2.3301495E 05
8.8000000E 00	-4.1701388E 05	-4.1701773E 05	-4.1701390E 05
9.6000000E 00	-7.0891059E 05	-7.0891518E 05	-7.0891060E 05

TABLE 2. NUMERICAL SOLUTION OF EQ. (3-11) (DX=3.2)

X	Y(EXACT)	Y(R-K)	Y(R-K-R)
0.0000000E 00	1.0000000E 00	1.0000000E 00	1.0000000E 00
3.2000000E 00	-8.1632700E 02	-9.3656376E 02	-8.1632703E 02
6.4000000E 00	-5.9431183E 04	-5.9940101E 04	-5.9431188E 04
9.6000000E 00	-7.0891059E 05	-7.1007577E 05	-7.0891071E 05

B. PERTURBATION METHOD ON CHEMICAL REACTIONS NEAR EQUILIBRIUM

In a gas-mixture flow field with variation of thermodynamic properties, the chemical reactions tend to maintain the gas mixture in chemical equilibrium. Let us represent the set of chemical reactions involving the species M_j ($j = 1, 2, \dots, s$) by



where $i = 1, 2, 3, \dots, r$ is the number of reactions. ν_{ij} and ν_{ij}^* are stoichiometric coefficients for reaction i . k'_{F_i} and k'_{B_i} are forward and reverse rate constant respectively. The k'_{F_i} can be calculated from the local vibrational equilibrium rate, $k_{F_{i\infty}}$

$$k'_{F_i} = k_{F_{i\infty}} \prod_{j=1}^s (\nu_{ij})^{A_{ij}} \quad (3-14)$$

where ν_{ij} is the vibrational coupling factor and A_{ij} may be 1 or 0, denoting which reaction i will be affected by the vibration-dissociation coupling process.

The net volumetric rate of production of species j from reaction i , Q'_{ij} , is given by

$$Q'_{ij} = \beta_{ij} \left[k'_{F_i} \rho^{\nu_{ij}} \prod_{\alpha=1}^s \gamma_{\alpha}^{\nu_{ij}} - k'_{B_i} \rho^{\nu_{ij}^*} \prod_{\alpha=1}^s \gamma_{\alpha}^{\nu_{ij}^*} \right] \frac{\text{moles}}{\text{cm}^3 \cdot \text{Sec}} \quad (3-15)$$

where $\nu_i' = \sum_{\alpha=1}^S \nu_{i\alpha}'$ and $\nu_i^* = \sum_{\alpha=1}^S \nu_{i\alpha}^*$

Since the equilibrium constant

$$K_i' = \frac{k_{F,i\infty}'}{k_{B,i\infty}'} = \frac{k_{F,i}'}{(V_j)^{\sum_{\alpha=1}^S \nu_{i\alpha}'} k_{B,i\infty}'} \quad (3-16)$$

Eq(3-15) may be rewritten

$$Q_{ij}' = \beta_{ij}' k_{F,i}' \left(\frac{p_i}{p} \right)^{\sum_{\alpha=1}^S \nu_{i\alpha}'} \gamma_{\alpha}'^{\nu_{i\alpha}'} \chi_i \quad (3-17)$$

where $\beta_{ij}' = \nu_{ij}^* - \nu_{ij}'$, $\beta_i = \sum_{j=1}^S \beta_{ij}'$

and

$$\chi_i = 1 - \frac{1}{(V_j)^{\sum_{\alpha=1}^S \nu_{i\alpha}'} K_i'} (p')^{\sum_{\alpha=1}^S \beta_{i\alpha}'} (\gamma_{\alpha}')^{\beta_{i\alpha}'} \quad (3-18)$$

The χ_i is referred to as the degree of nonequilibrium of the i^{th} reaction. When the reaction i is in equilibrium $\chi_i = 0$. Otherwise, the χ_i will have the values between 1 and $-\infty$, excluding 0.

In the flow field of gas mixture, if the chemical reaction time is slower than the time characterizing the variation of thermodynamic properties in the flow, the concentrations of the species near equilibrium usually oscillate around the equilibrium values as shown in Figure 1.

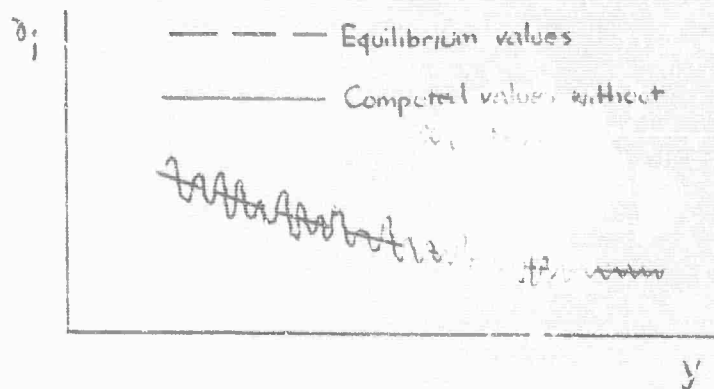


Figure 1 Variation of Species Concentration Near Equilibrium

Using the Runge-Kutta method to calculate the concentration of species along the streamwise distance, y , the integration step size has to be smaller than the distance between two adjacent relative maximum or relative minimum points in the actual concentration curve to attain the desired accuracy. Near equilibrium, the distance in the y -direction between two relative extremes of the species concentration curve is usually very small and derivatives become large. Hence, the computer running time for a flow field at near-equilibrium is very long. The χ_i - variation of those reactions near equilibrium is very slow along the streamwise distance, y . Also, in this region the values of χ_i are very near zero. As an approximation, we can set χ_i equal to zero. Since the χ_i vs y curve is nearly flat, the results of this approximation will not deviate too much from the actual values, but with this approximation the integration step size can be much larger. The scheme of this perturbation method is as follows.

Two specified bandwidths, 2δ and 2ϵ are placed around $\chi_i = 0$ in the χ_i vs y space where δ is less than ϵ . We keep checking the χ_i -values at every integration interval. If the previous χ_i -value is not zero and the current $|\chi_i|$ calculated from its algebraic definition (see Eq. (3-18)) is smaller than δ , we set $\chi_i = 0$ which makes the $Q_{ij} = 0$ from Eq. (3-17). This term is omitted from the summation in Eq. (2-8) for integration, and the reaction it represents is temporarily and artificially frozen. If the $|\chi_i|$ calculated from Eq. (3-18) is equal to or greater than δ , then the currently integrated χ_i -value is used. If the previous χ_i -value is zero, we set $\chi_i = 0$ for $|\chi_i|$ less than ϵ . Otherwise, the calculated value from Eq. (3-18) is used as the updating χ_i -value. Thus, those nonzero $|\chi_i|$ -values less than ϵ never enter the integration. When the chemical reaction moves away from equilibrium, calculated values from Eq. (3-18) are used until its $|\chi_i|$ exceeds ϵ . In this way, instead of using one

δ -band, we can avoid the rapid fluctuation of the $|\chi_i|$ -values from 0 to the calculated value greater than δ and vice versa. This χ_i -test is an empirical method. Hence, the bandwidths for δ and ϵ should be determined by experiments for each particular class of problem. Experience is gained rapidly and fixed δ or ϵ values work for a wide range of conditions. Far less high-speed memory storage is required for this method than for a separate analytical perturbation code. With the streamtube program described above, most of the available cells are occupied so that the artificial perturbation χ_i -test represents a significant advantage. It probably obviates the need for a chain program modification in this case.

IV PROGRAM DESCRIPTION

This program is written in FORTRAN IV for an IBM 7040-44 digital computer. This program requires approximately 25,000 memory cells and at least two magnetic tapes for the normal outputs. The system at GM DRL is composed of a 4K central system monitor, with 32K of core storage, 8 magnetic-tape drives and a 1611 card read-punch unit on channel A. With specific sense switch settings, we can select optional outputs on tape #2 and tape #3. The sense switch settings and tape requirements are as follows:

A. Sense Switch Settings:

- #1 - optional output on tape #2 of species thermodynamic and chemical variables
- #3 - optional output on tape #6 of the array of coefficients c of the set of simultaneous differential equations
- #4 - optional output on tape #6 of the derivatives of boundary condition function with respect to y .
- #5 - optional output on tape #3 of the molar rate of production, α_{ij}
- #6 - include χ_i ($i = 1 \dots N$) in each result paragraph on tape #4.

B. Tape Requirements:

Output tape #2 (optional output), Tape #2 is written only when sense switch #1 is on and $DUMP_{IND}$ (input variable) is set equal to 1 or 4. A paragraph of species thermodynamic and chemical variable is printed for the first Runge-Kutta

step, $DUMP_{IND}=1$, or all four Runge-Kutta steps,
 $DUMP_{IND}=4$ in an interval.

Output Tape #3 (optional output) - Tape #3 is written only when sense switch #5 is on. For each success step a paragraph of the molar rate of production of species ($j=C+1, \dots, S$) from all reactions i , Q_{ij} , is printed out.

Output tape #4 (normal output) - A paragraph of results for each printing interval is printed out.

Output tape #6 (normal output) - Error messages of failing test are printed out on tape #6. When sense switch 4 is on, the derivatives of boundary condition with respect to y are printed on tape #6.

The chemical system of this streamtube program has a capability to read in forty reversible chemical reactions ($i=1, 2, 3, \dots, r$) and twenty chemical species ($j=1, 2, 3, \dots, S$). The species are subdivided into independent species, the elements ($k=1, 2, 3, \dots, C$), and dependent species ($j=C+1, \dots, S$). In the dependent species we can have some diatomic species ($j=f+1, \dots, g$) for which vibration nonequilibrium are considered.

With this program, we can put more than one set of streamtube calculations in one run. There is one data card right after the first set

of data cards to tell how many sets of data we have for this run; we put all the sets of data right after it in one deck and run them. When it is finished the system will put "end of file" on every tape used. For every streamtube there are two normal termination conditions. One is $Y_n \leq Y'_{stop}/L'$ (input variable). The other one is $T_n \leq T_{min}$. (input variable TEST₂). At normal termination, the following line will be printed on tape #6:

RUL (run no.)

COMPLETED

There are two possible error terminations. One is due to the zero determinant encountered in the set of simultaneous linear differential equations. The other one is due to $\Delta y_n \leq \Delta y_{min}$. If either occurs, the error messages are printed out on tape #6 and the system will go on to process the next set of data. If the error termination occurs in the final set of data, this run is finished.

REFERENCES

1. Paul V. Marrone, "Inviscid, Nonequilibrium Flow Behind Bow and Normal Shock Waves, Part I: General Analysis and Numerical Solutions," CAL Rept. No. QM-1626-A-12(I), May 1963
2. Leonard J. Garr, and Paul V. Marrone, "Inviscid, Nonequilibrium Flow Behind Bow and Normal Shock Waves, Part II: The IBM 704 Computer Programs," CAL Rept. No. QM-1626-A-12(II), May 1963
3. P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, John Wiley & Sons, Inc., 1962
4. E. B. Hildebrand, Introduction to Numerical Analysis, McGraw-Hill Co., 1956

APPENDIX A
INPUT FORMATS

	<u>Page</u>
A. First Five Cards	A-1
B. Species Thermodynamics	A-3
C. Chemical Reaction System	A-3
D. α j1. Matrix	A-4
E. Initial Condition	A-4
F. Boundary Condition	A-5
G. Boundary Condition Polynomial	A-5
H. Boundary Condition Table	A-6
I. Number of Data Sets	A-6

APPENDIX A INPUT FORMATS

The order and format of input data cards are given in detail below.

A. First Five Cards

First Card - Format (A6, 8I3, 2E10.4)

IRUN	run number
S	number of species in the gas mixture
r	number of chemical reactions in the gas mixture
f, g	define vibrational nonequilibrium group of species
C	number of elements
B. C.	boundary condition, 1 \rightarrow U, 2 \rightarrow ρ , 3 \rightarrow p & 4 \rightarrow A
E. E _{INL}	electronic excitation of the species, 0 \rightarrow no, 1 \rightarrow yes
DUMP _{IND}	optional output of i^{th} step of Runge-Kutta, 0 \rightarrow none, 1 \rightarrow first, 4 \rightarrow all
$\Delta y'_{\text{start}}$	starting interval, in cm
y'_{stop}	value of y' to stop computation, in cm

Second Card Format (3E9.5)

p'_0	reference pressure in dyne/cm ²
ρ'_0	reference density in gm/m ³
T'_0	reference temperature in °K
Blank	available for further information
A'_0	initial cross-sectional area of streamtube in cm ²

U'_0	reference velocity in cm/sec
MW'_0	reference molecular weight, in gm/mole
L'	reference length in cm, usually one cm

Third Card Format (5E14.7)

TEST ₁	number of output result paragraphs per page
TEST ₂	value of temperature that terminates the computation
TEST ₃	percentage change in T allowed for each interval
TEST ₄	halfwidth of outer χ_i test band, usually = 0.1
TEST ₅	half width of inner χ_i test band, usually = 0.05

Fourth Card Format (5E14.7)

TEST ₆	blank
TEST ₇	blank
TEST ₈	blank
TEST ₉	first value of y to be printed after the initial value of y nondimensional, $y_p = y'_p / L'$
TEST ₁₀	printing interval, $\Delta y_p = \Delta y'_p / L'$

Fifth Card Format (5E14.7)

TEST ₁₁	number of successful steps before replacing Δy_{11} by $C \Delta y_n$
TEST ₁₂	C , step size increment factor
TEST ₁₃	blank
TEST ₁₄	minimum Δy allowed
TEST ₁₅	maximum Δy allowed

B. Species Thermodynamics

There are four cards for each of the $j = 1, \dots, s$ species

First Card Format (6E12.6)

n_j number of atoms per molecule, one or two
 b_j constant for chemical potential calculation
 θ_{vj} characteristic vibrational temperature, in $^{\circ}\text{K}$
 h_j° heat of formation, calories per mole
 N_j number of vibrational levels for harmonic oscillator cut off at dissociation energy

Blank available for further species description

Second Card Format (4E12.6, 4A6)

$\tau_{aj}, \tau_{bj}, \tau_{cj}, \tau_{dj}$ describe vibrational relaxation time for species in the $f + 1 \rightarrow g$ group;

$$\tau_j' p' = \tau_{aj} (\tau')^{\tau_{bj}} \exp \left[\frac{\tau_{cj}}{(\tau')^{\tau_{dj}}} \right] \frac{\text{dynes}}{\text{cm}^2} - \text{sec}$$

SPECJK j^{th} species identification

Third Card Format (12, 8F2.0)

M_{jl} number of electronic levels of j^{th} species, $l = 1, \dots, 8$
 g_{jl} degeneracy of each l^{th} electronic level

Fourth Card Format (8E9.5)

E_{jl} energy levels of electronic states in cal/molecule

C. Chemical Reaction System

There are two cards for each $i = 1, \dots, r$ reaction.

First Card Format (3F2.0, 20F1.0, 20F 1.0, 20F 1.0)

W_i, Z_i, D_i denotes which ϕ_j' relation to use

ν_{ij}^* right-hand side stoichiometric coefficient of species j on reaction i

ν_{ij} left-hand side stoichiometric coefficient of species j on reaction i

A_{ij} denotes which species is involved with vibrational dissociation coupling of reaction i , 1 \rightarrow yes, 0 \rightarrow no

Second Card format (I4, 4E14.7, 2A6)

KFIIND denotes direction of input reaction rate constant,
0 \rightarrow forward, 1 \rightarrow backward

$A_{k, \infty}, B_{k, \infty},$
 $C_{k, \infty}, D_{k, \infty}$ reaction rate constant

$$k_{B_{k, \infty}} \text{ or } k_{F_{k, \infty}} = A_{k, \infty} (T')^{B_{k, \infty}} \exp \left[- \frac{C_{k, \infty}}{(T')^{D_{k, \infty}}} \right] \frac{\text{cm}^6}{\text{mole}^2 \cdot \text{sec}} \text{ or } \frac{\text{cm}^3}{\text{mole} \cdot \text{sec}}$$

SPECIK identification of i^{th} reaction

D. α_{jk} Matrix Format (20F2.0)

Need one card for each $k = 1, \dots, C$ element to describe all species by the elements

E. Initial Condition

First Card Format (5E14.7)

One card for every five of $f + 1 \rightarrow g$ species. If $f = g$, omit this card.

ϵ_j vibrational energy of j^{th} species

Second Card Format (5E14.7)

One card for every i e species

γ_j concentration of j^{th} species, nondimensional $\gamma_j = \gamma_j' * MW_0'$

Third Card Format (5E14.7)

y starting distance $y = y'/L'$
 T starting temperature $T = T'/T_0'$
 p starting pressure $p = p'/(p_0' U_0'^2)$
 ρ starting density $\rho = \rho'/\rho_0'$
 U starting velocity $U = U'/U_0'$

Fourth Card Format (5E14.7)

MW starting molecular weight of the mixture, $MW = \frac{MW'}{MW_0'}$

F. Boundary Condition

The boundary condition can be read in either in tabular or polynomial forms.

First Card Format (2I3)

NOR number of the polynomial or number of cards for the table
IPOT polynomial or table indicator, 0 \rightarrow polynomial,
1 \rightarrow table.

G. Boundary Condition Polynomial (IPOT = 0)

One set for each polynomial

First Card Format (I3, 2E14.7)

IOOP order of the polynomial
REST starting coordinate, y , of this polynomial
REND ending coordinate, y , of this polynomial

Second Card Format (5E14.7)

One card for every five coefficients

COP coefficient of polynomial, starting with the lowest
 degree coefficient

H. Boundary Condition Table (IPOT=1) Format (ZE14.7)

One card for each TI

TY coordinate value

TP corresponding value of the boundary condition function

I. Number of Data Sets Format (I2)

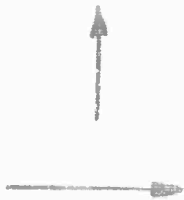
ISTOP number of data sets for this run. Only need to
 put one "number of data sets" card behind the first
 set of data.

APPENDIX B
FLOW DIAGRAM

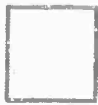
		<u>Page</u>
S01	MAIN PROGRAM	B-3
S02	Subroutine START	B-6
S03	Subroutine READIN	B-7
S04	Subroutine WRITE 1	B-8
S05	Subroutine WRITE 2	B-9
S06	Subroutine MATRIX	B-10
S07A	Subroutine DER	B-11
S07B	PREFERENTIAL FLOW I, II, III	B-15
S08	Subroutine SUB1	B-18
S09	Subroutine SUB2	B-19
S10	Subroutine SUB4	B-20
S11	Subroutine SUB5	B-21
S12	Subroutine SUB6	B-22
S13	Subroutine TEST1	B-23
S14	Subroutine TEST2	B-24
S15	Subroutine TEST3	B-25
S16	Subroutine OVRFLW	B-26
S17	Subroutine LCOUNT	B-27
S18	Subroutine STOP	B-28
S19	Subroutine SIMSOL	B-29
S20	Function DERT	B-30
S21	Function YINT	B-31

APPENDIX B FLOW DIAGRAM

The detailed flow diagram of all subroutines used in the stream-tube program are presented in this section. The definitions of the symbols used in the flow diagram are given below.



Direction of flow



The beginning or ending of a program



Connector. To connect parts of a flow chart.



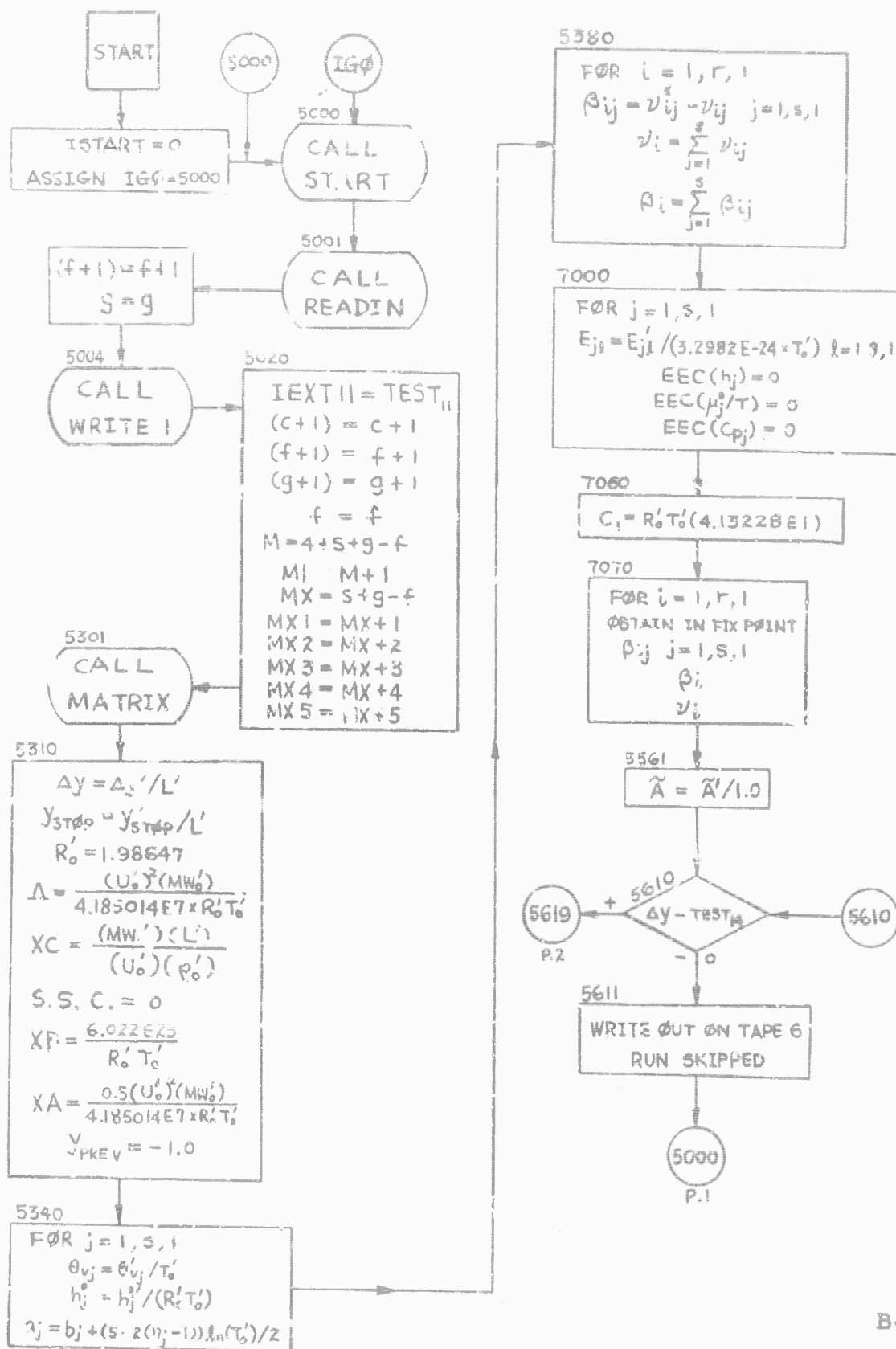
Decision function. Branch to one of two or more alternate paths is possible

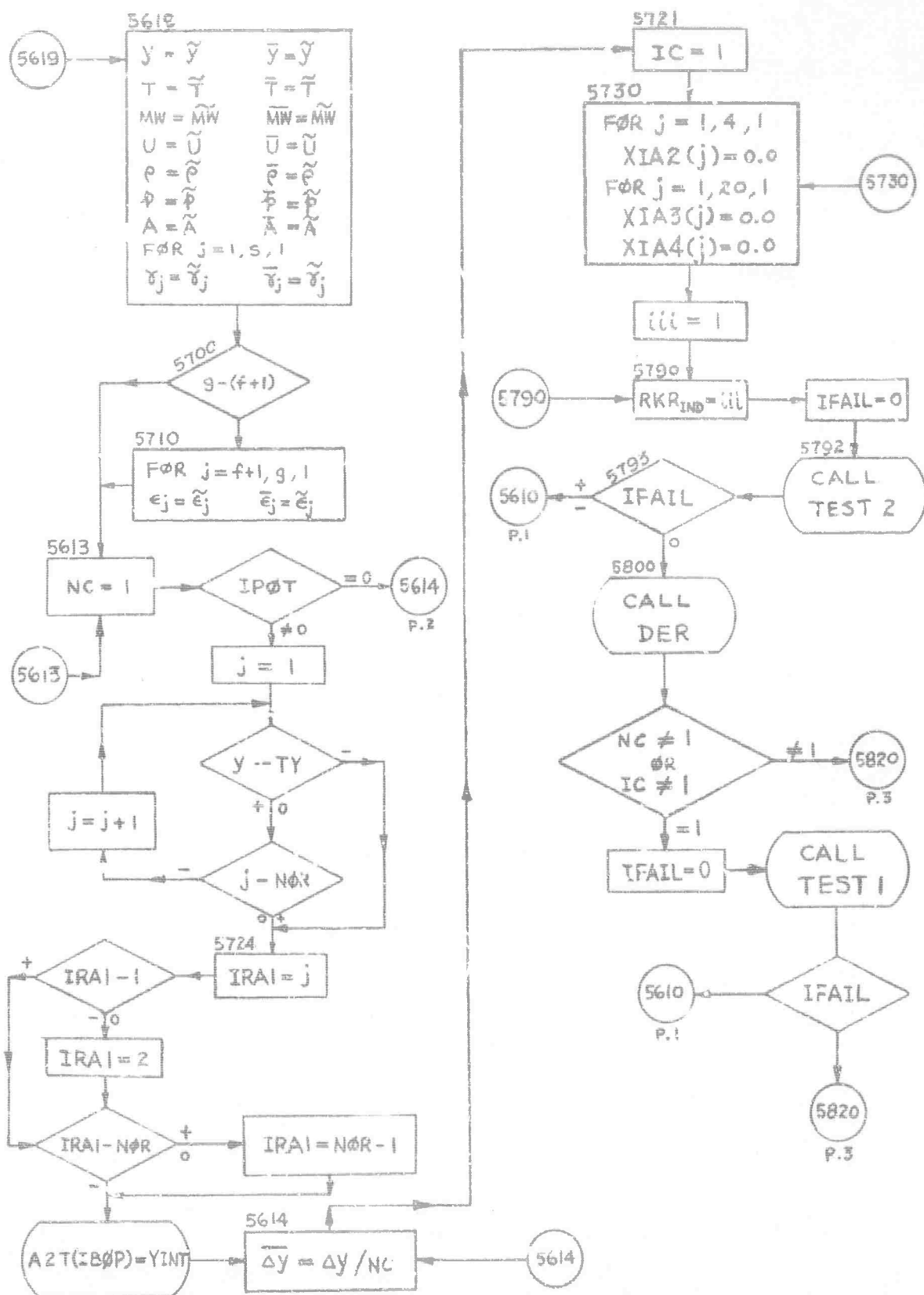


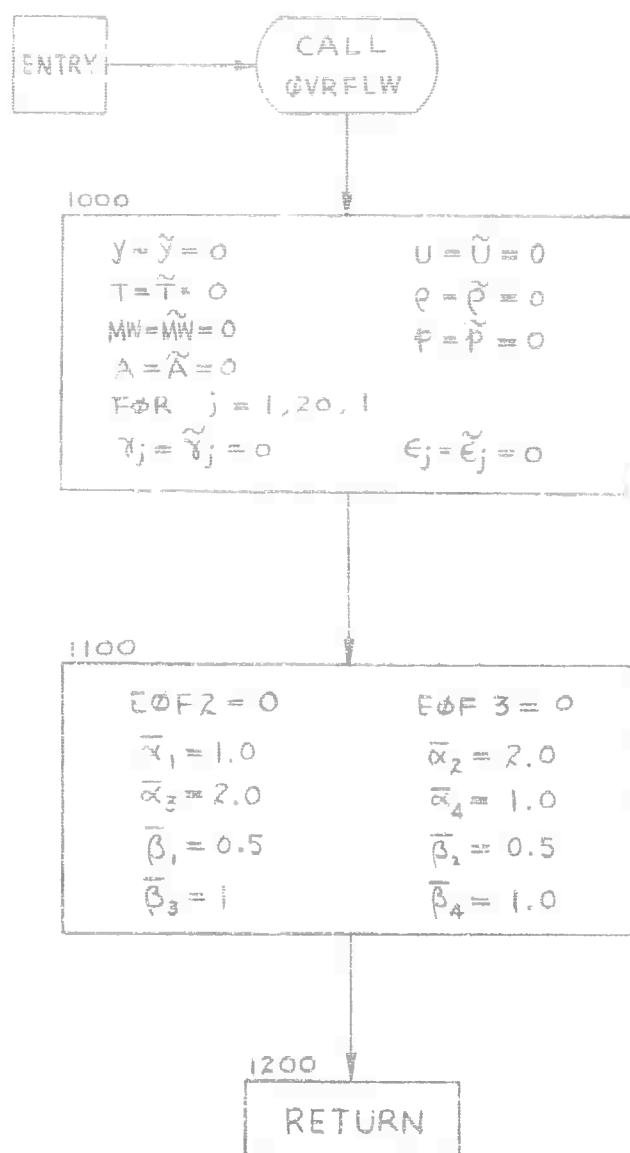
Processing or operational box. Descriptive
or working block of instructions.

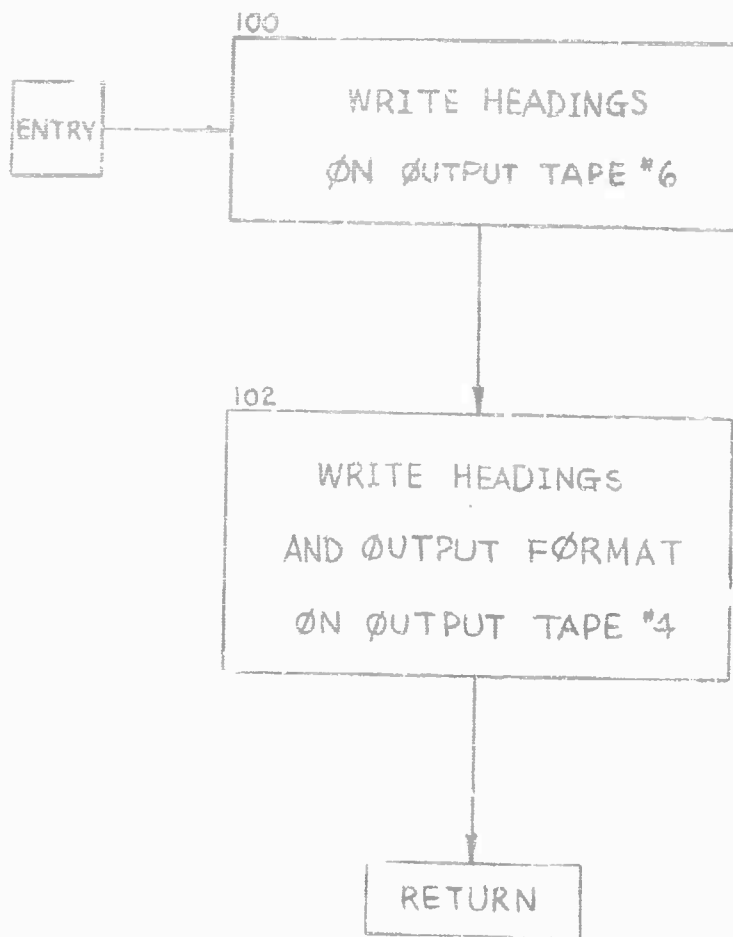


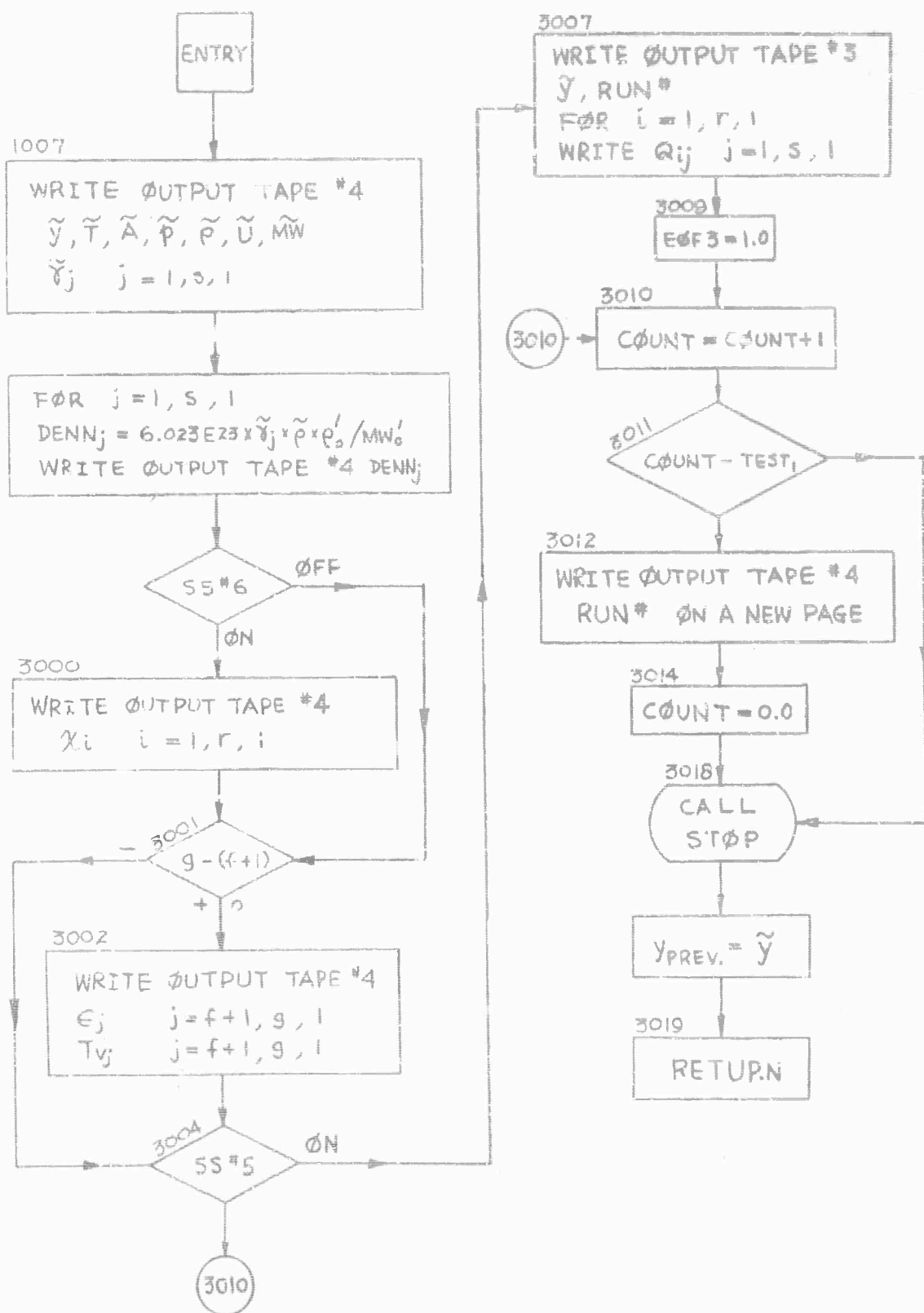
Call box. To call a library or closed subroutine.

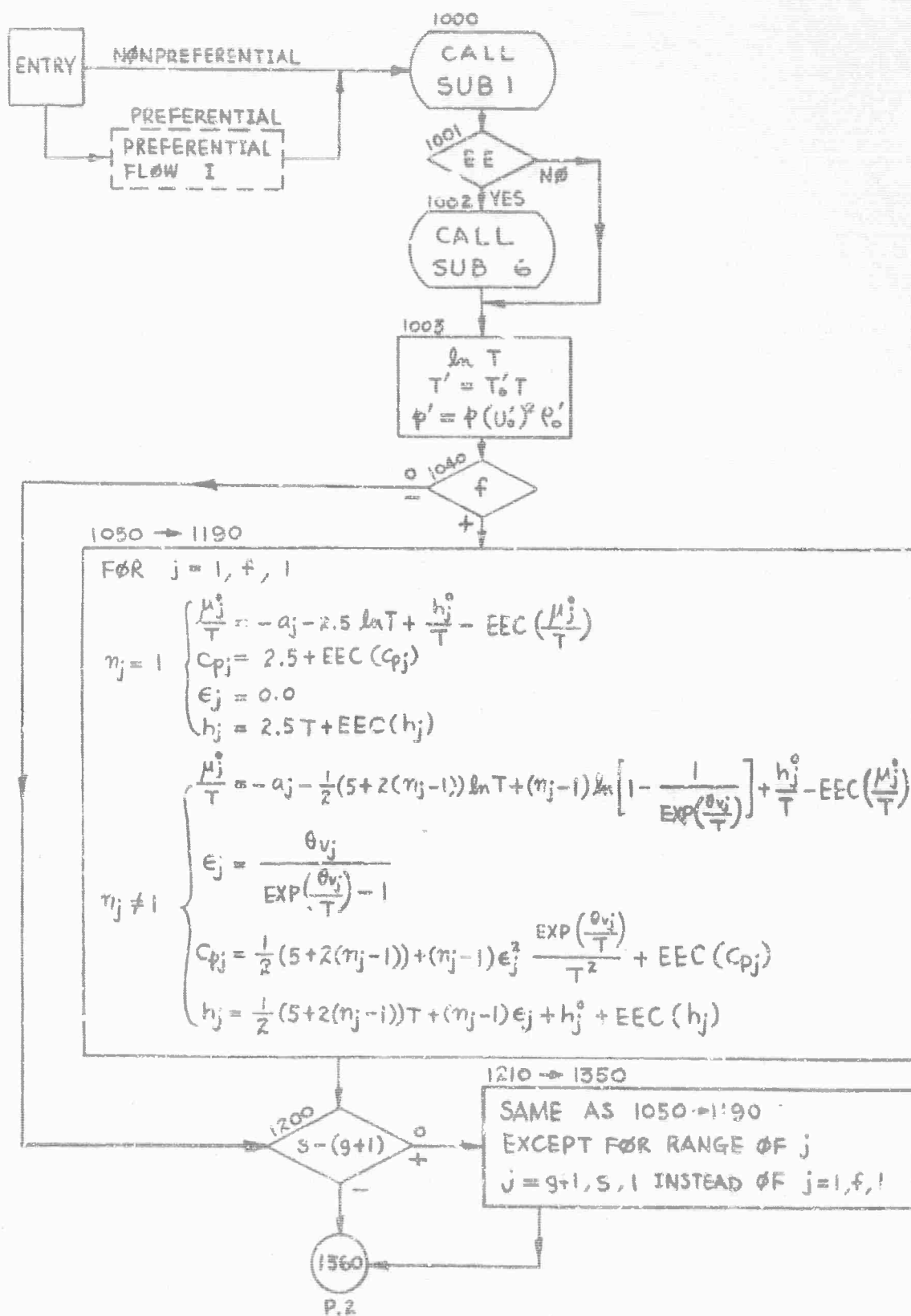


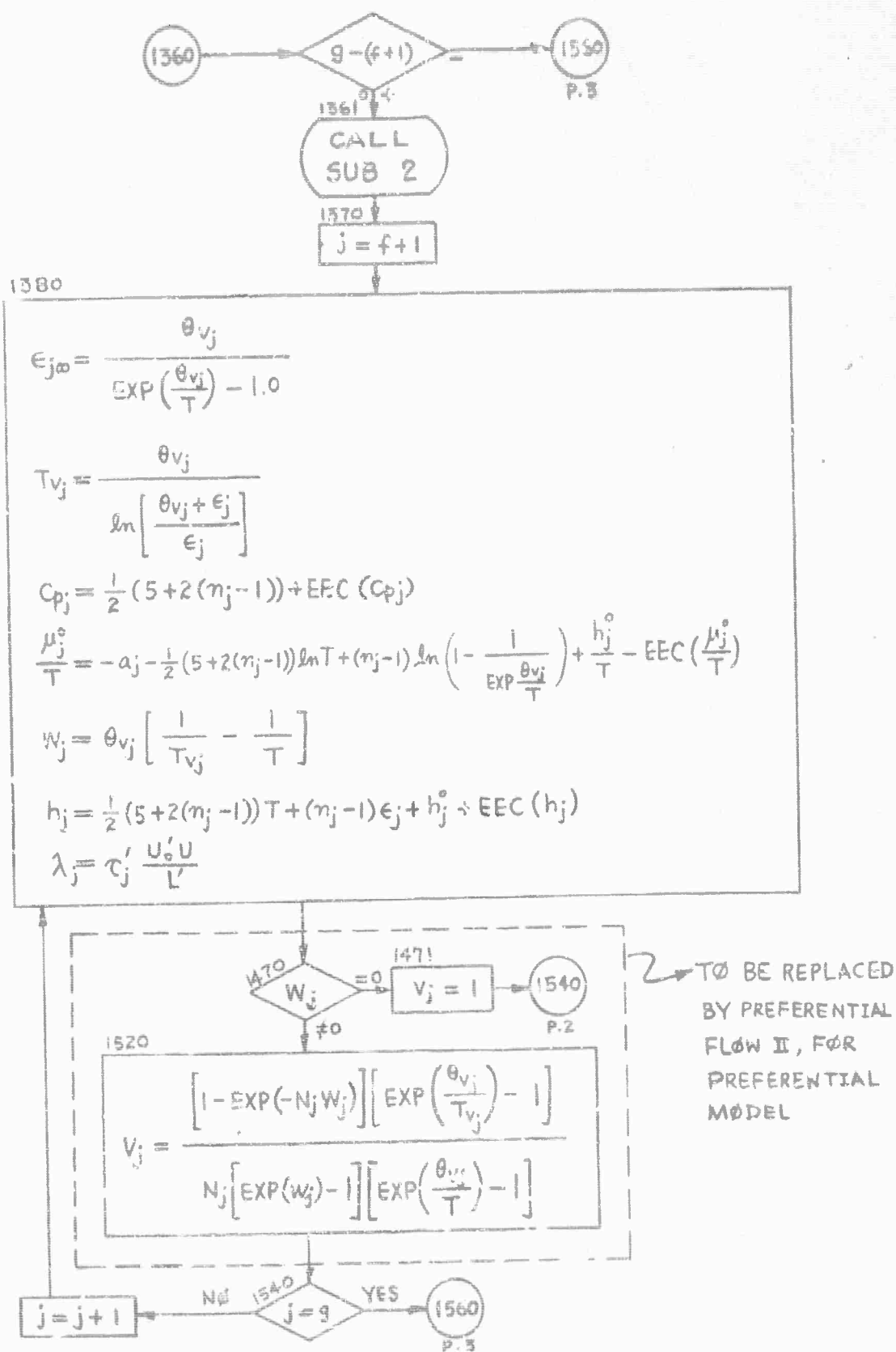


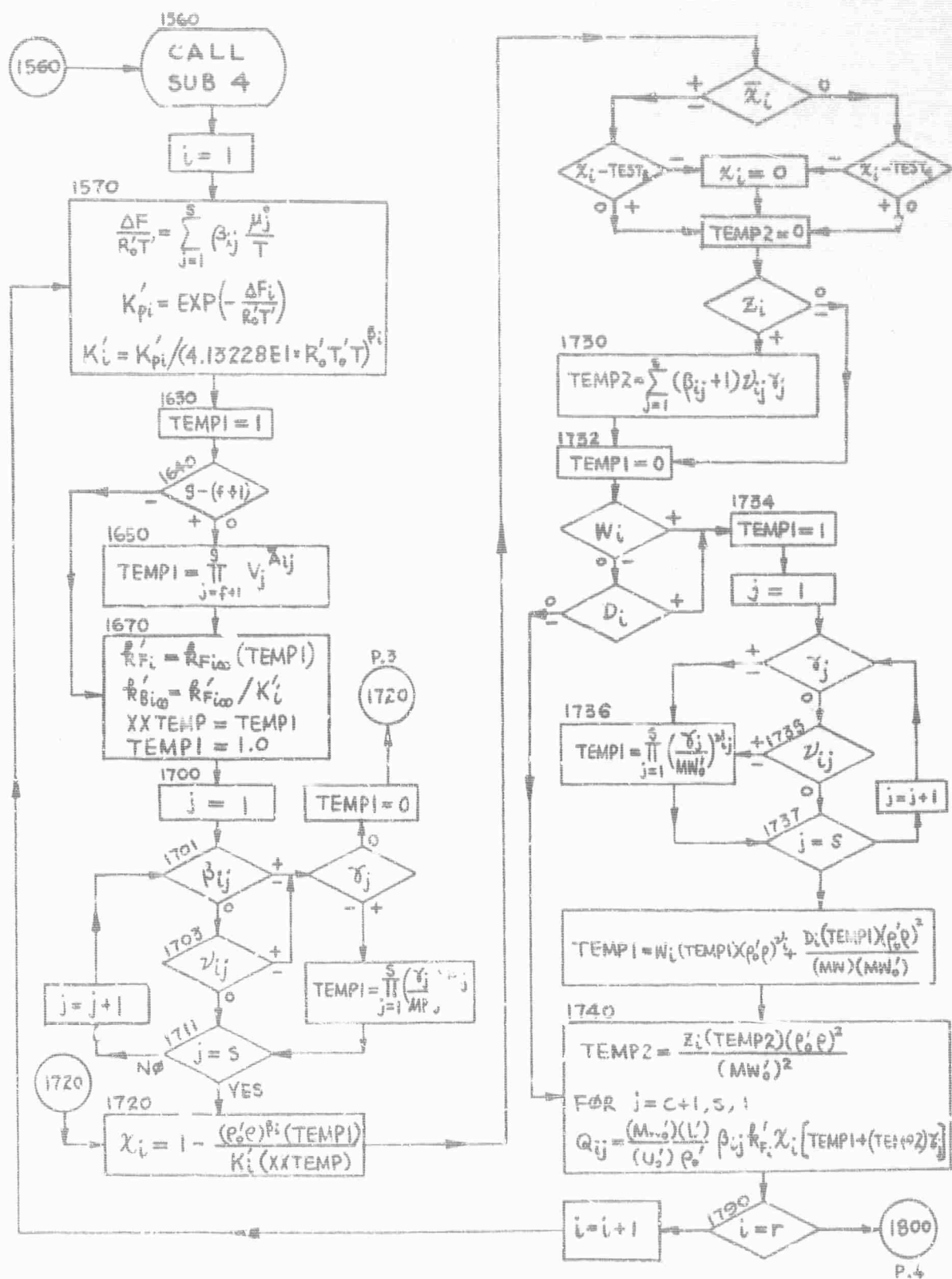


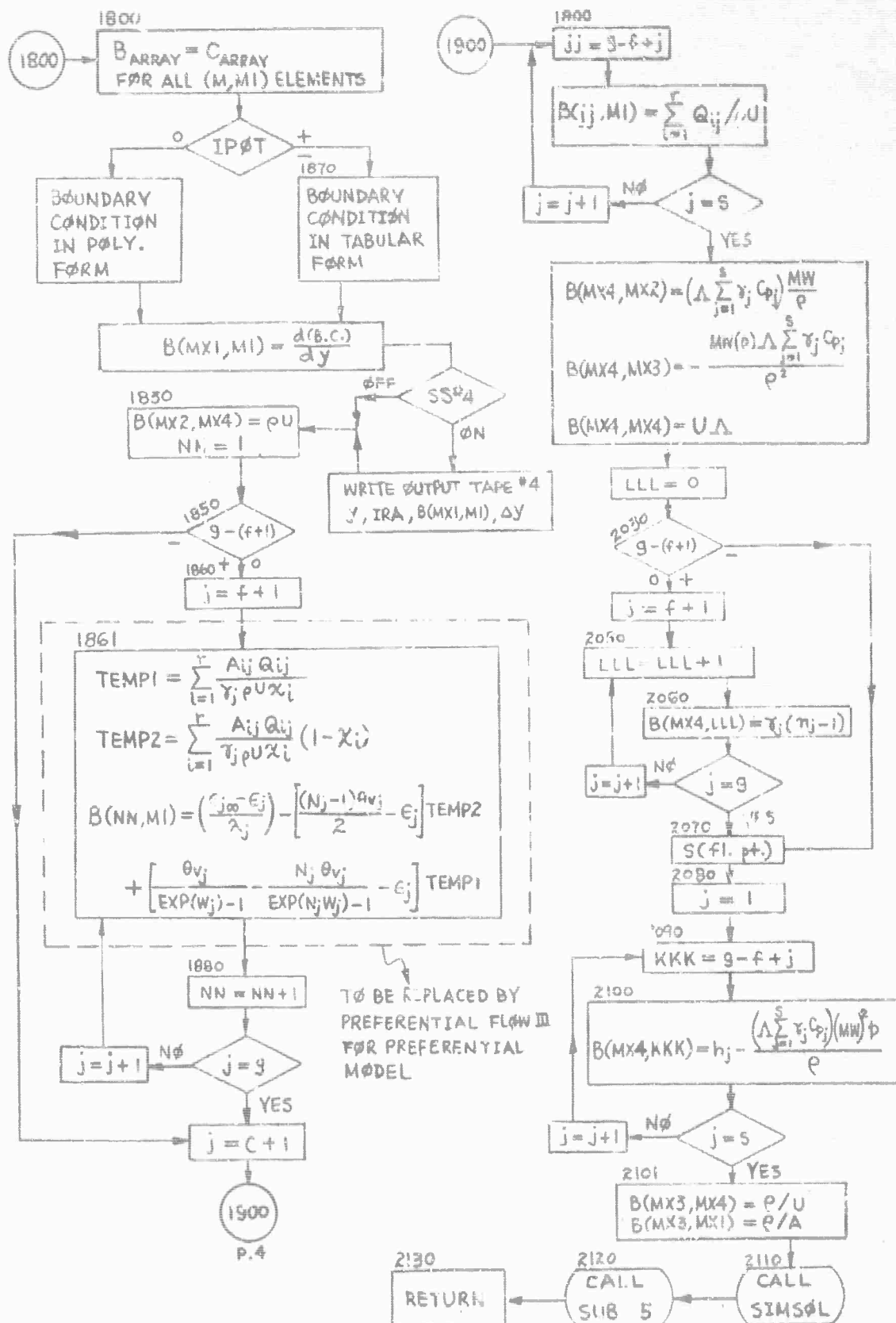


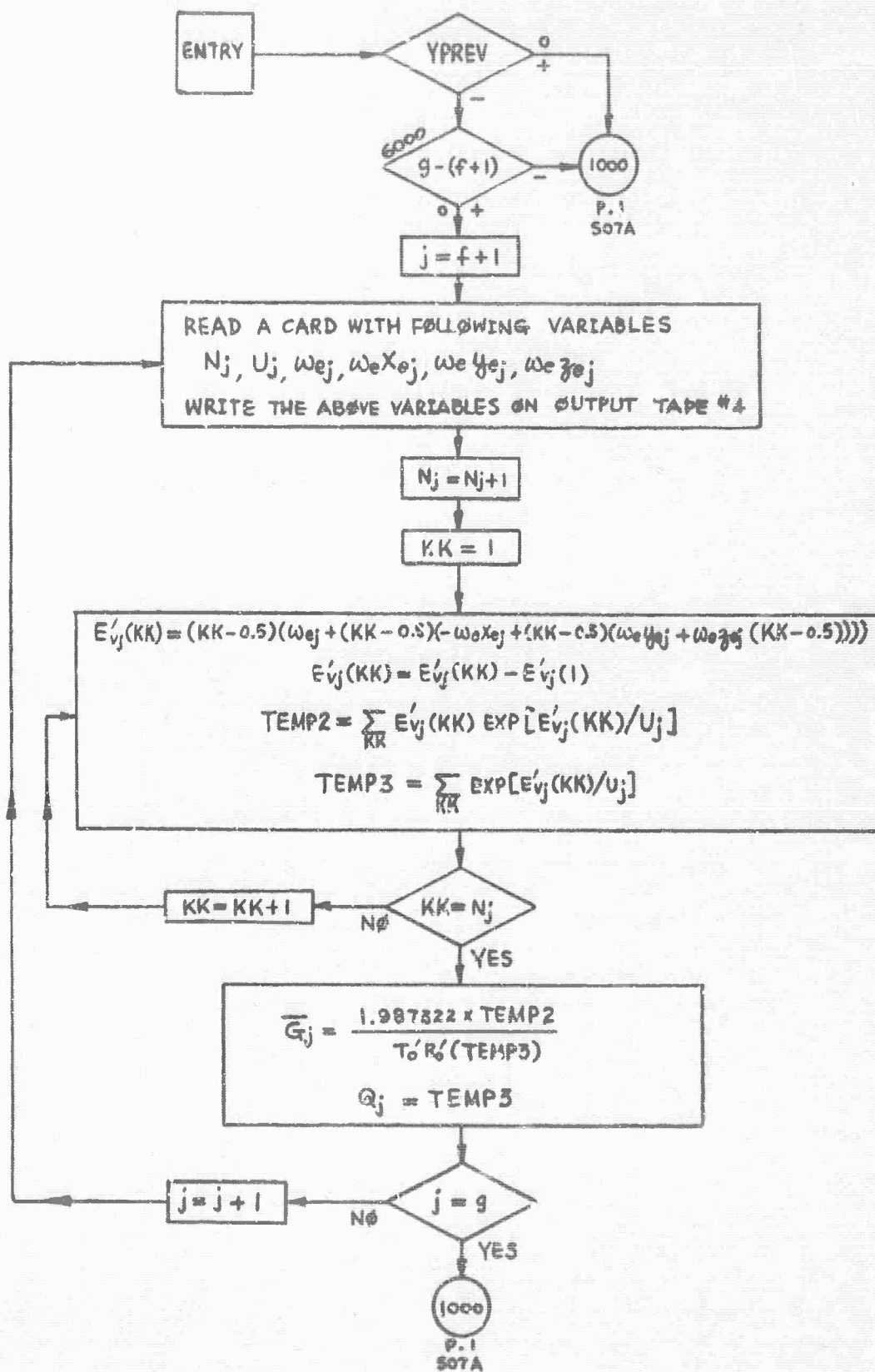


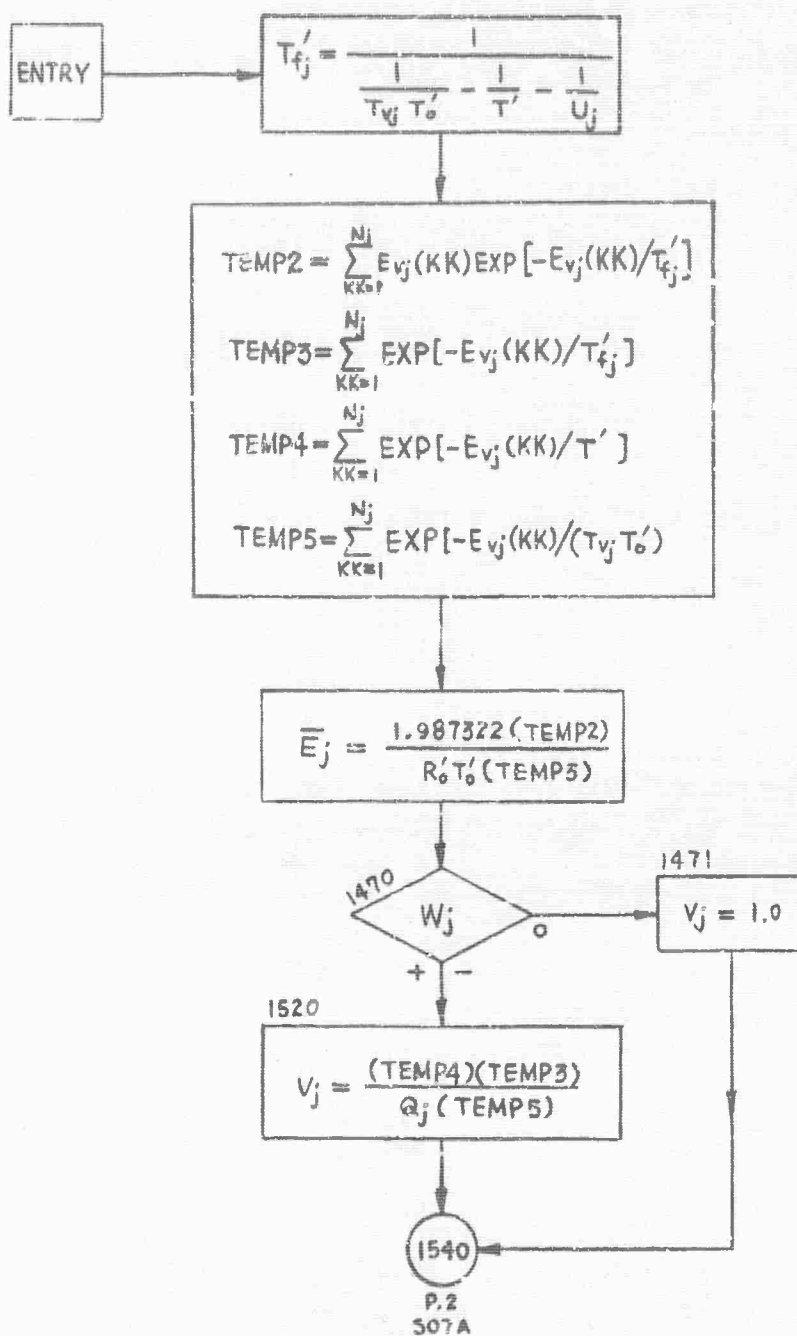


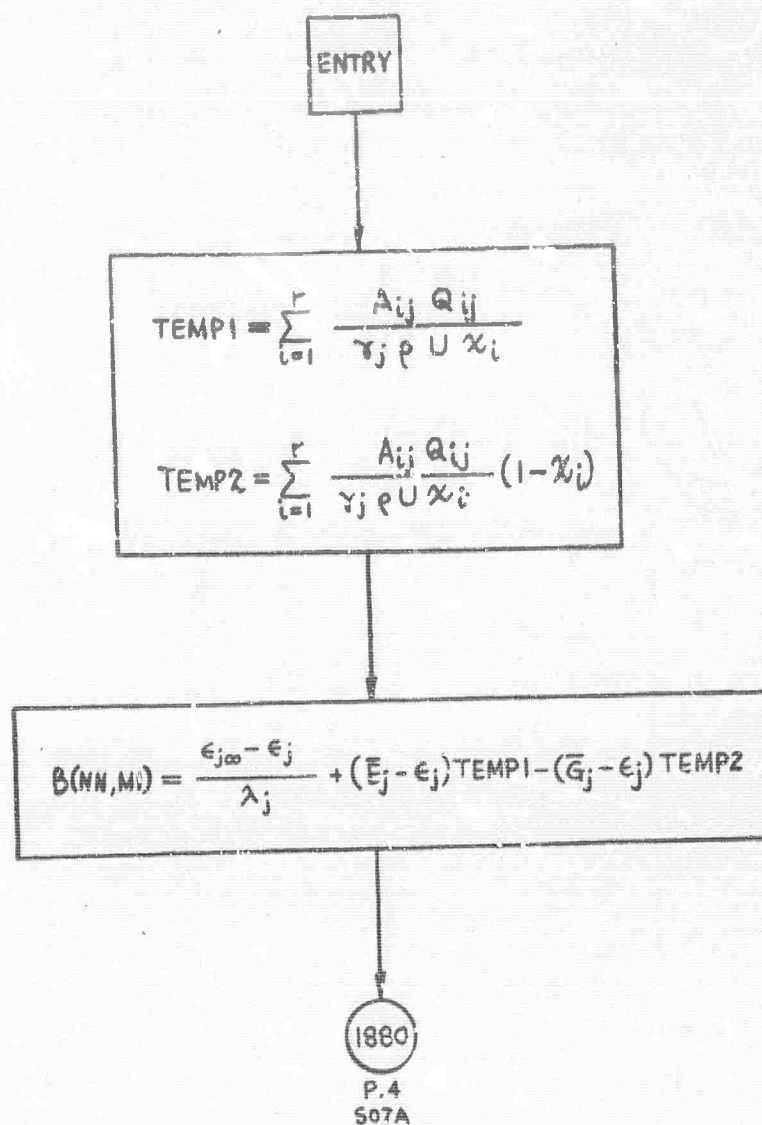


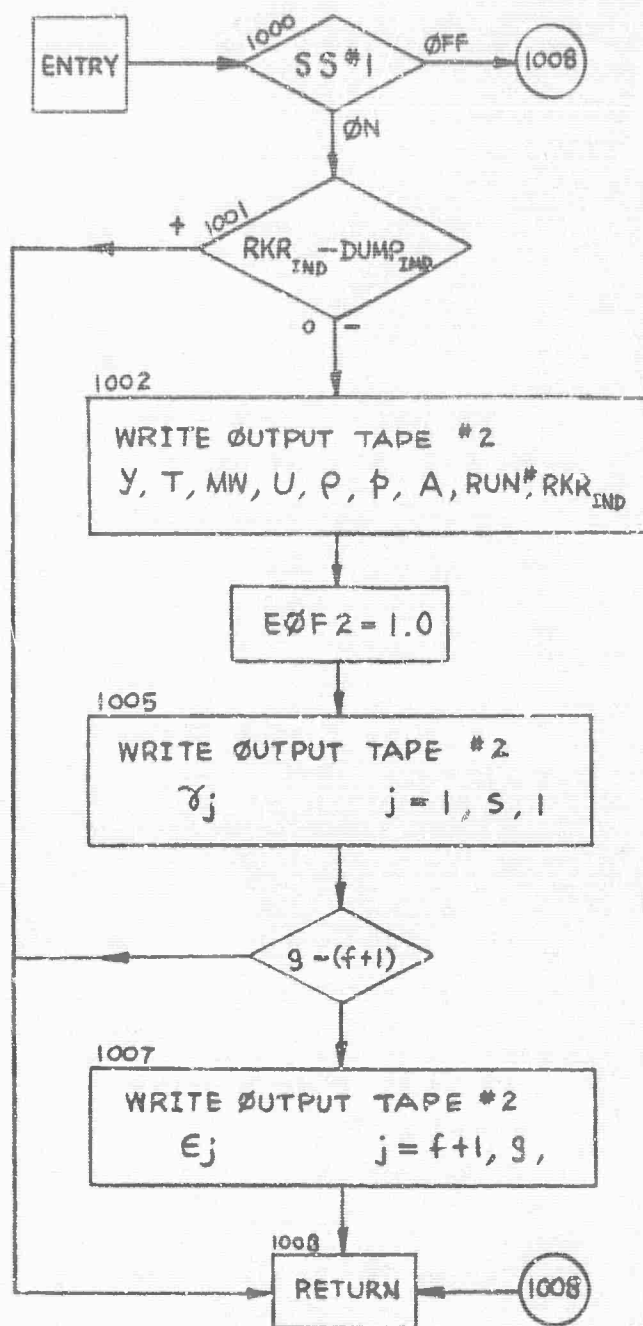


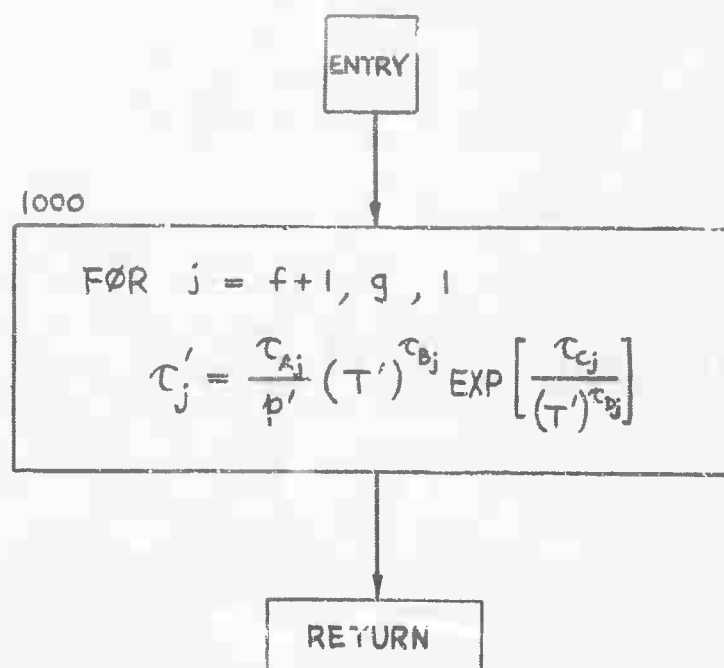


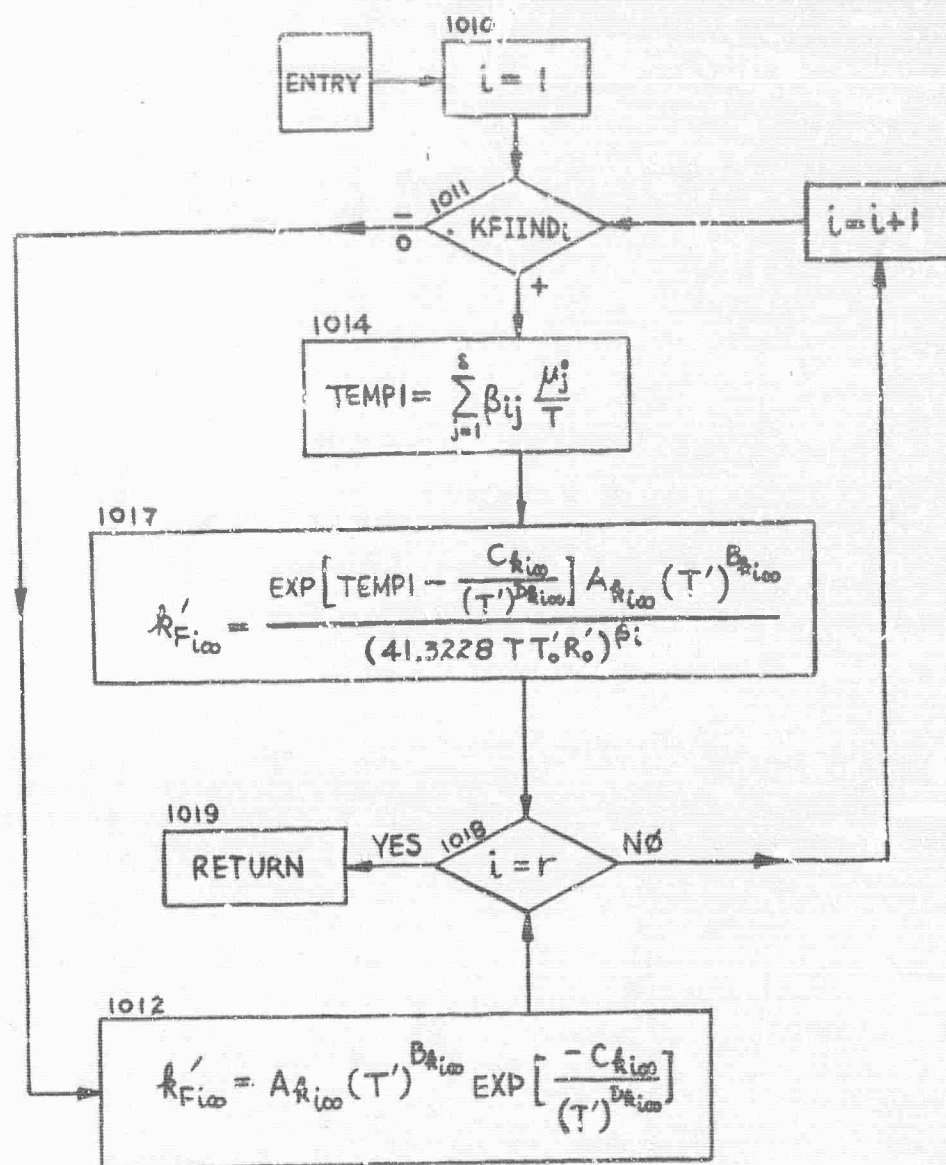


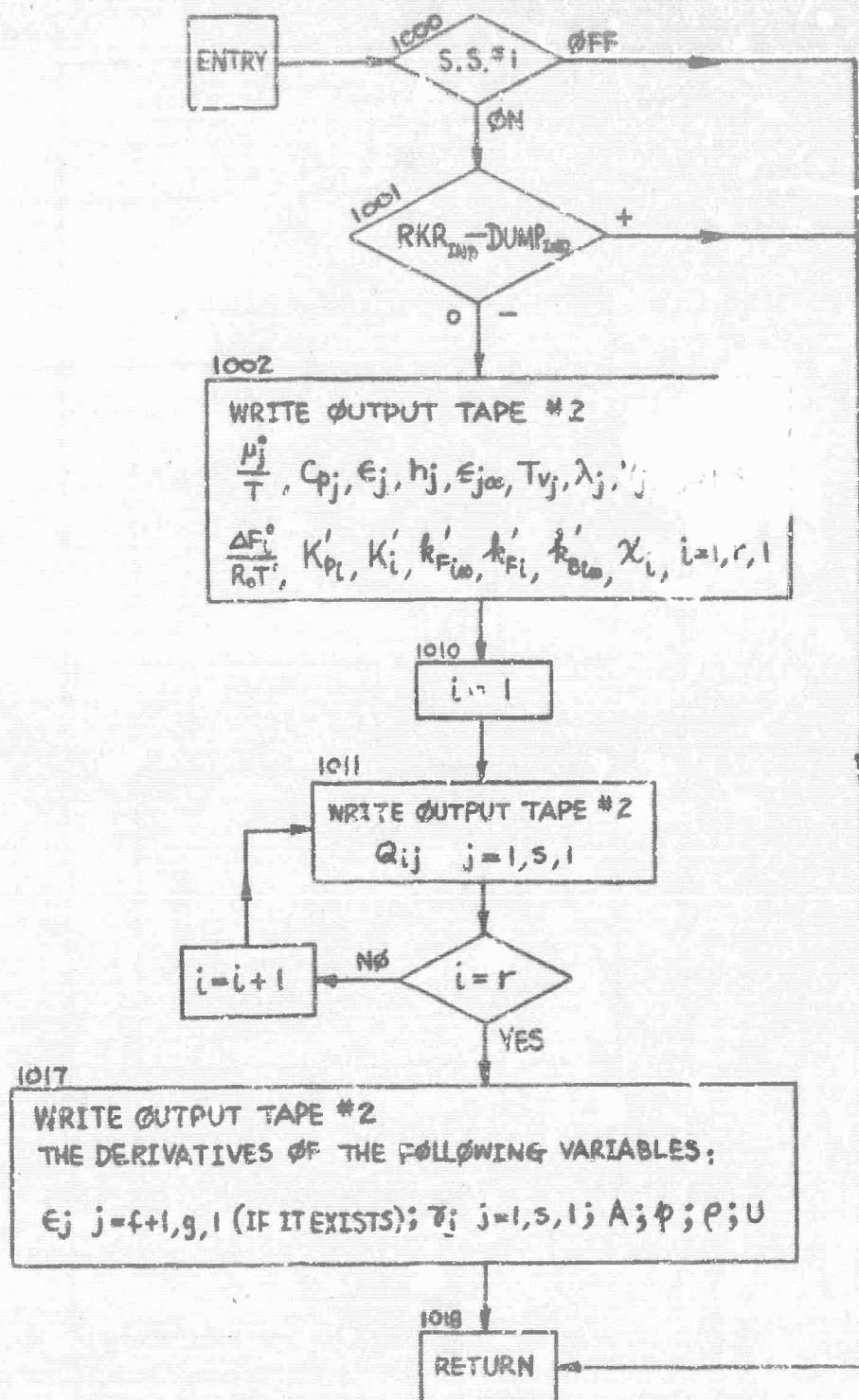


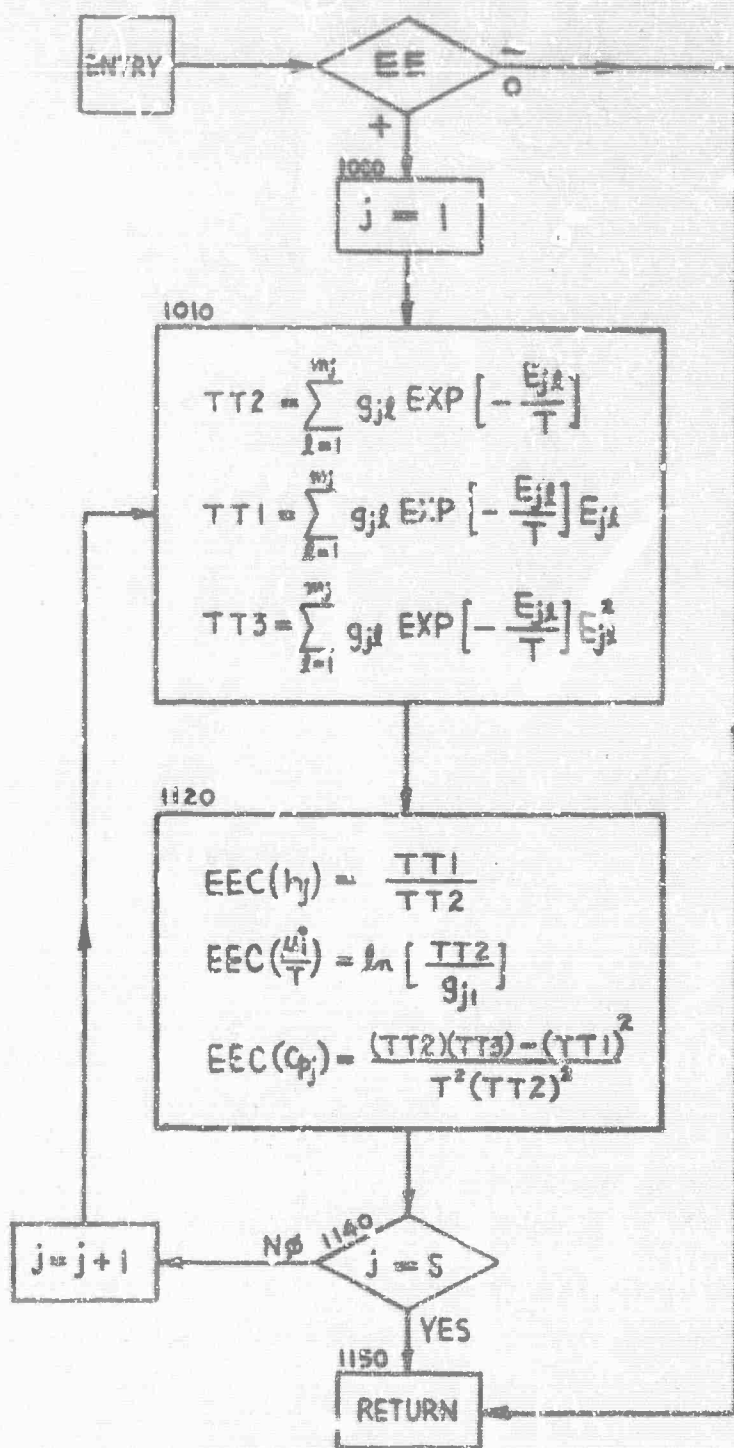


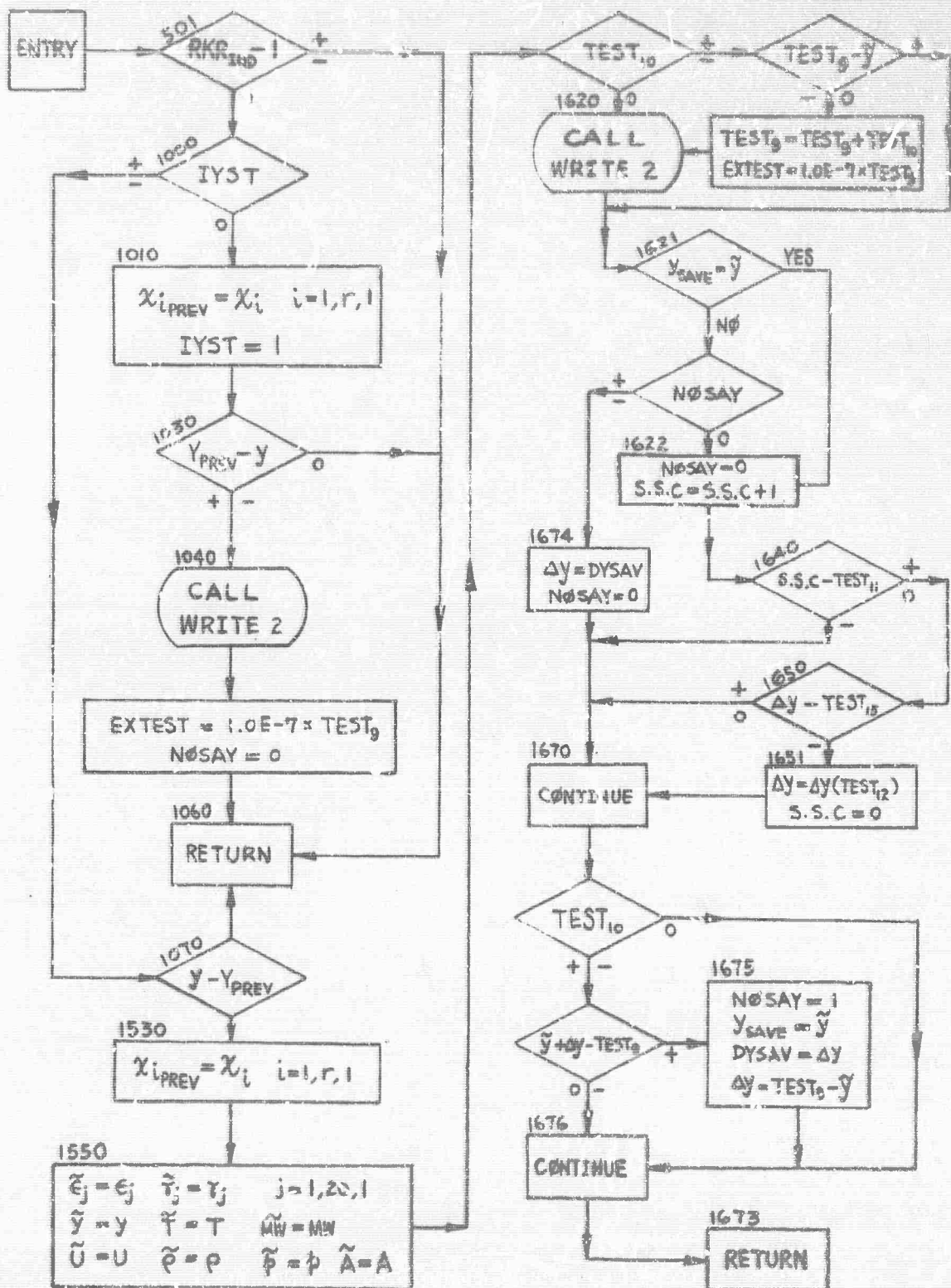


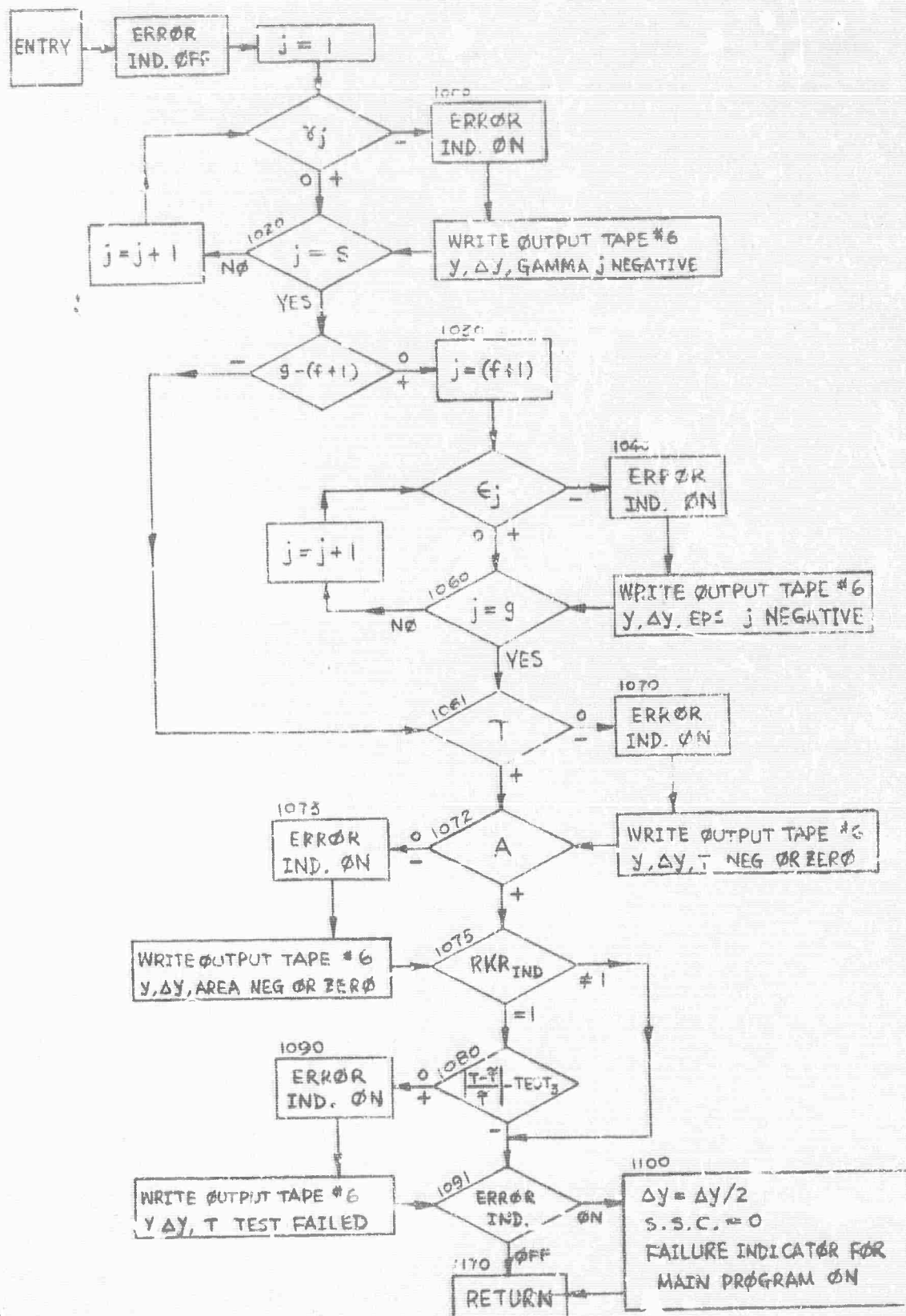


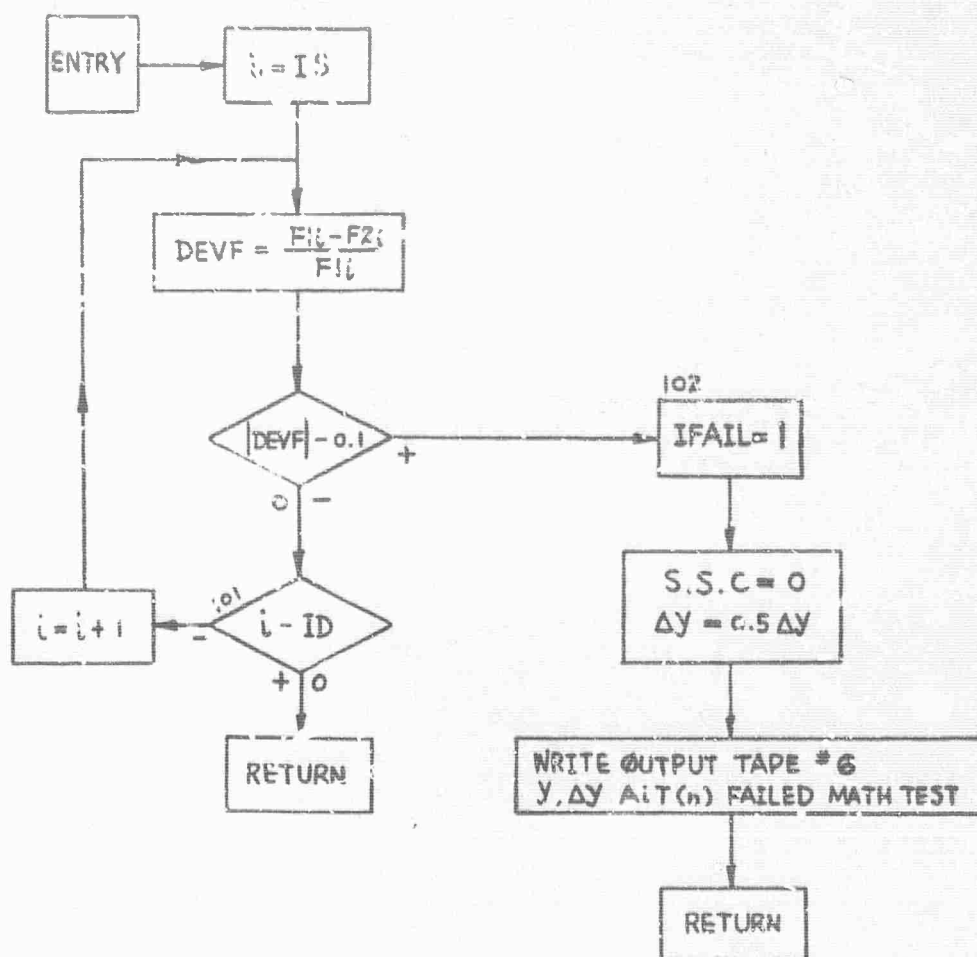








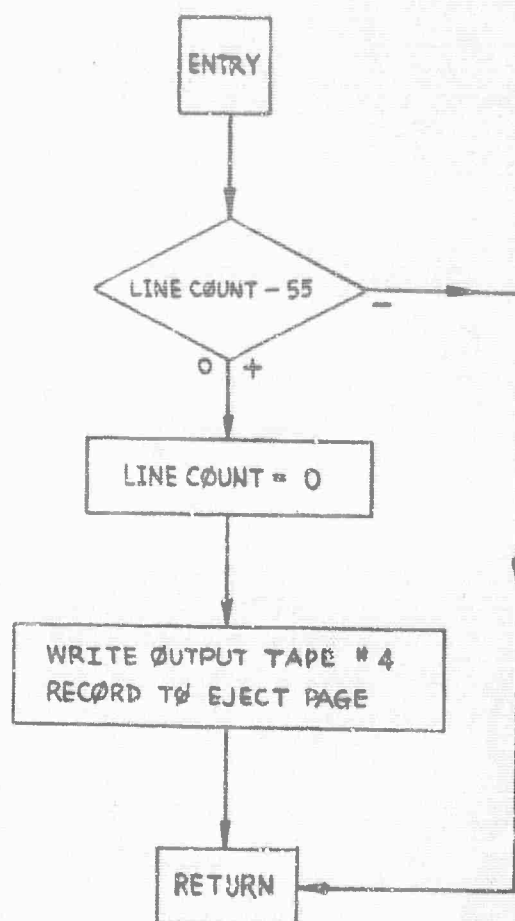


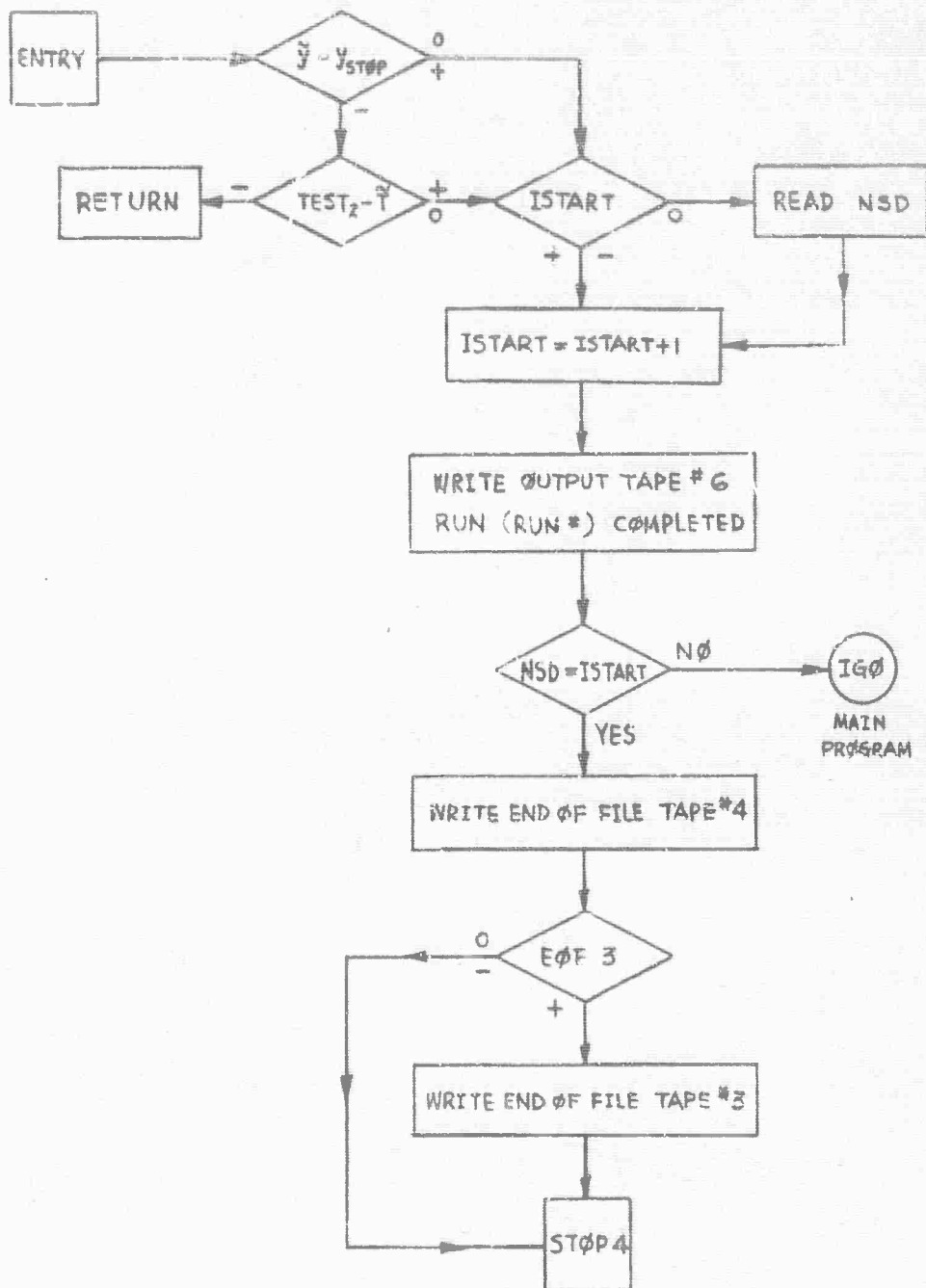


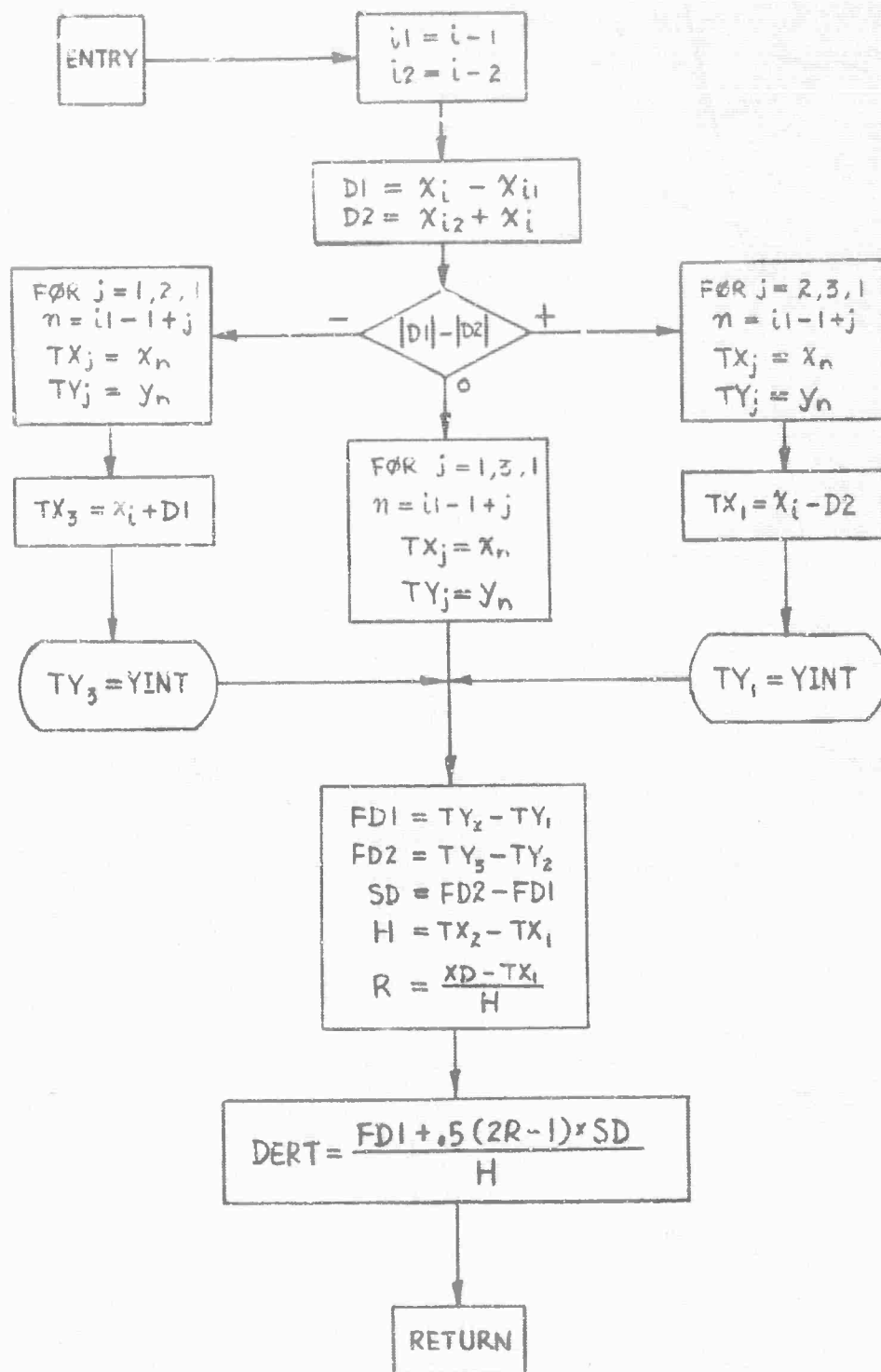
SUBROUTINE OVRFLW

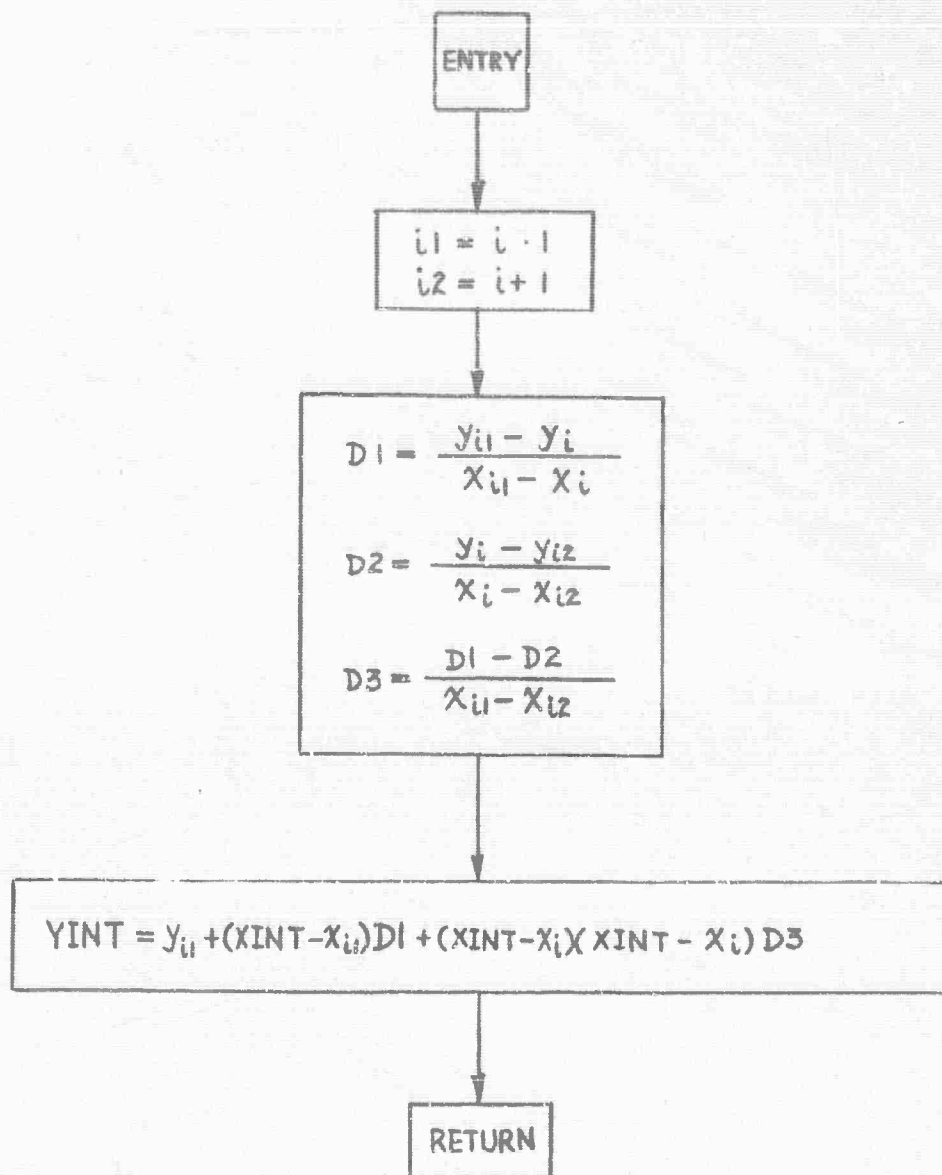


SUBROUTINE LCOUNT









APPENDIX C
PROGRAM LISTING AND SYMBOLS

1. LIST OF FORTRAN SYMBOLS	C-1
2. PROGRAM LISTINGS	C-9
S01 MAIN PROGRAM	C-12
S02 Subroutine START	C-16
S03 Subroutine READIN	C-17
S04 Subroutine WRITE 1	C-21
S05 Subroutine WRITE 2	C-24
S06 Subroutine MATRIX	C-25
S07A Subroutine DER	C-26
S07B PREFERENTIAL FLOW I, II, III	C-30
S08 Subroutine SUB1	C-35
S09 Subroutine SUB2	C-36
S10 Subroutine SUB4	C-37
S11 Subroutine SUB5	C-38
S12 Subroutine SUB6	C-39
S13 Subroutine TEST1	C-40
S14 Subroutine TEST2	C-41
S15 Subroutine TEST3	C-42
S16 Subroutine OVRFLW	C-43
S17 Subroutine LCOUNT	C-44
S18 Subroutine STOP	C-45
S19 Subroutine SIMSOL	C-46
S20 Function DERT	C-48
S21 Function YINT	C-49
S22 Comman and Dimension Statements	C-50

APPENDIX C PROGRAM LISTING AND SYMBOLS

1. List of Fortran Symbols

The Fortran Symbols listed here are given in alphabetical order.

The corresponding report symbols and their meaning are also given in this section.

A1I(3)	$A1I(1) = \bar{y}$ $A1I(2) = \bar{T}$ $A1I(3) = \bar{MW}$	INITIAL CONDITION FOR A R-K-R INTEGRATION INTERVAL
A1IC(3)	$A1IC(1) = \bar{y}$ $A1IC(2) = \bar{T}$ $A1IC(3) = \bar{MW}$	UPDATED INITIAL CONDITION IN A R-K-R INTERVAL
A1T(3)	$A1T(1) = y$ $A1T(2) = T$ $A1T(3) = MW$	UPDATED VALUE IN A R-K-R INTEGRATION INTERVAL
A1T1(3)	$A1T1(1) = y_1$ $A1T1(2) = T_1$ $A1T1(3) = MW_1$	UPDATED VALUE IN A R-K-R INTERVAL, FULL INT. STEP SIZE
A2I(4)	$A2I(1) = \bar{U}$ $A2I(2) = \bar{p}$ $A2I(3) = \bar{p}$ $A2I(4) = \bar{A}$	INITIAL CONDITION FOR A R-K-R INTEGRATION INTERVAL
A2IC(4)	$A2IC(1) = \bar{U}$ $A2IC(2) = \bar{p}$ $A2IC(3) = \bar{p}$ $A2IC(4) = \bar{A}$	UPDATED INITIAL CONDITION IN A R-K-R INTERVAL
A2T(4)	$A2T(1) = U$ $A2T(2) = p$ $A2T(3) = p$ $A2T(4) = A$	UPDATED VALUE IN A R-K-R INTEGRATION INTERVAL
A2T1(4)	$A2T1(1) = U_1$ $A2T1(2) = p_1$ $A2T1(3) = p_1$ $A2T1(4) = A_1$	UPDATED VALUE IN A R-K-R INTERVAL, FULL INT. STEP SIZE
A3I(J)	$\tilde{x}_j \quad j=1, \dots, S$	INITIAL CONDITION FOR A R-K-R INTEGRATION INTERVAL
A3IC(J)	$\tilde{x}_j \quad j=1, \dots, S$	UPDATED INITIAL CONDITION IN A R-K-R INTERVAL
A3T(J)	$\tilde{x}_j \quad j=1, \dots, S$	UPDATED VALUE IN A R-K-R INTEGRATION INTERVAL

A3T1(J)	$v_j \quad j = 1, \dots, 9$	UPDATED VALUE IN A R-K-R INTERVAL, FULL INT. STEP SIZE
A41(J)	$\tilde{e}_j \quad j = f+1, \dots, 9$	INITIAL CONDITION FOR A R-K-R INTEGRATION INTERVAL
A41C(J)	$\tilde{e}_j \quad j = f+1, \dots, 9$	UPDATED INITIAL CONDITION IN A R-K-R INTERVAL
A4T(J)	$e_j \quad j = f+1, \dots, 9$	UPDATED VALUE IN A R-K-R INTEGRATION INTERVAL
A4T1(J)	$e_j \quad j = f+1, \dots, 9$	UPDATED VALUE IN A R-K-R INTERVAL, FULL INT. STEP SIZE
AKFI(1)	$A_{fi\infty}$	FORWARD OR BACKWARD RATE CONSTANT COEFFICIENT
ALPHA(4)		RUNGE-KUTTA-RICHARDSON COEFFICIENT
ALP1J(J,K)	α_{jk}	NUMBER OF k ATOMS IN j SPECIES
B(45,45)		ARRAY OF CONSTANT COEFFICIENTS OF THE SET OF SIMUL. EQS
BETA(4)		RUNGE-KUTTA-RICHARDSON COEFFICIENT
BETA1(1)	β_i	$\beta_i = \sum_{j=1}^5 \beta_{ij}$
BETA1J(1,J)	β_{ij}	$\beta_{ij} = z_{ij}^* - \lambda_{ij}$
BKFI(1)	$B_{fi\infty}$	FORWARD OR BACKWARD RATE CONSTANT COEFFICIENT
C(45,45)		ARRAY OF COEFFICIENTS OF THE SET OF SIMULTANEOUS EQS.
CAOP	\tilde{A}'	INITIAL CROSS-SECTIONAL AREA
CA1J(1,J)	A_{ij}	VIBRATION-DISSOCIATION COUPLING INDICATOR
CCPJ(J)	C_{pj}	SPECIFIC HEAT
CDI(1)	D_i	DEMOTES WHICH SPECIES PRODUCTION RELATION TO USE
CEALP(J,L)	E_{jL}	DIMENSIONLESS ENERGY OF ELECTRONIC LEVEL L
CEALPX(J,L)	E'_{jL}	ENERGY OF ELECTRONIC LEVEL L
CHI1(1)	χ_i	DEGREE OF NONEQUILIBRIUM OF REACTION i
CHI1T(1)	$\bar{\chi}_i$	DEGREE OF NONEQUILIBRIUM OF REACTION i ON PREVIOUS STEP
CKFI(1)	$C_{fi\infty}$	FORWARD OR BACKWARD RATE CONSTANT COEFFICIENT
CKI(1)	K'_i	EQUILIBRIUM CONSTANT OF REACTION i
CKPI(1)	K''_i	EQUILIBRIUM CONSTANT OF REACTION i
CLP	L'	REFERENCE LENGTH
CNWDOP	MW'_0	REFERENCE MOLECULAR WEIGHT

CNJ(J)	N_j	NUMBER OF VIBRATIONAL LEVELS
CON1	C_1	$41.3228 * CR0 * CTOP$
COP(K,N)	$C\phi P_n$	COEFFICIENT OF $X^{*(N-1)}$ IN B. C. POLYNOMIAL
COUNT		LINE COUNTER
CP0P	p'_c	REFERENCE PRESSURE
CPP	p'	PRESSURE
COIJ(1,J)	Q_{ij}	DIMENSIONLESS RATE OF PRODUCTION
CR0	R'_c	GAS CONSTANT. 1.98647 (CAL/MOLE-°K)
CTOP	T'_c	REFERENCE TEMPERATURE
CTP	T'	TEMPERATURE
CTVJ(J)	T_{vj}	VIBRATIONAL TEMPERATURE
CU0P	U'_c	REFERENCE VELOCITY
CVJ(J)	V_j	VIBRATIONAL COUPLING FACTOR
CWI(1)	w_i	DENOTES WHICH SPECIES PRODUCTION RELATION TO USE
CWJ(J)	w_j	$\theta_{vj} [1/T_{vj} - 1/T]$
CXEJ(J)	X_{ej}	NOT USED. AVAILABLE FOR FURTHER SPECIES DESCRIPTION
CZI(1)	z_i	DENOTES WHICH SPECIES PRODUCTION RELATION TO USE
DELY	Δy	RUNGE-KUTTA-RICHARDSON FULL STEP SIZE
DELYC	$\overline{\Delta y}$	RUNGE-KUTTA-RICHARDSON UPDATED STEP SIZE
DELYP	$\Delta y'_{START}$	STARTING VALUE OF STEP SIZE FOR R-K-R INTEGRATION
DENN(J)	ND_j	NUMBER DENSITY OF SPECIES j
DEVF		RESIDUE IN A R-K-R INTERVAL
DF10(1)	$\Delta F^\circ_c / (R'_c T')$	CHANGE IN STANDARD FREE ENERGY FOR REACTION i
DKFI(1)	$D_{k_{i00}}$	FORWARD OR BACKWARD RATE CONSTANT COEFFICIENT
EECCPJ(J)	$EEC(C_{pj})$	ELECTRONIC EXCITATION CONTRIBUTION FOR CCPJ(J)
EECHJ(J)	$EEC(h_j)$	ELECTRONIC EXCITATION CONTRIBUTION FOR SPJ(J)
EECNUJ(J)	$EEC(\mu_j^\circ / T)$	ELECTRONIC EXCITATION CONTRIBUTION FOR SUJOUT(J)

EOF 2		INDICATOR FOR WRITING END OF FILE ON TAPE 2
EOF 3		INDICATOR FOR WRITING END OF FILE ON TAPE 3
EPSJ(J)	ϵ_j	VIBRATIONAL ENERGY
EPSJN(J)	ϵ_{j0}	VIBRATIONAL ENERGY AT LOCAL TRANSLATIONAL TEMPERATURE
ETAJ(J)	η_j	NUMBER OF ATOMS PER MOLECULE
EXTRAJ(J)	TEST J	TEST PARAMETER, $j = 1, \dots, 15$
GAMMAO		NOT USED, AVAILABLE FOR FURTHER SPECIFICATION
GJOP(J)	γ_{j0}	REFERENCE SPECIES CONCENTRATION
IBETAI(I)	β_i	$\beta_i = \sum_{j=1}^5 \beta_{ij}$
IBETIJ(I,J)	β_{ij}	$\beta_{ij} = \nu_{ij}^* - \nu_{ij}$
IBOP		INDICATOR FOR BOUNDARY CONDITION
ICATJ(I,J)	A_{ij}	VIBRATION-DISSOCIATION COUPLING INDICATOR
ICON		NOT USED, AVAILABLE FOR FURTHER SPECIFICATION
IDELXC	S.S.C.	SUCCESSFUL STEP COUNTER
IDUMP	DUMP _{IND}	INDICATOR FOR OPTIONAL OUTPUT ON TAPE 2
TEXT 11		TEST 11 (FIXED POINT)
IFAIL		MAIN PROGRAM FAILURE INDICATOR
IGO		STARTING ADDRESS OF THIS PROGRAM
IISF	f	DEFINE VIBRATIONAL NONEQUILIBRIUM RANGE
IISFPI	f+1	ISF + 1
IISG	g	DEFINE VIBRATIONAL NONEQUILIBRIUM RANGE
INDSUM	E.E.	INDICATOR FOR ELECTRONIC EXCITATION
IOOP(K)		ORDER OF BOUNDARY CONDITION POLYNOMIAL
IPOT		INDICATOR FOR B, C, POLYNOMIAL OR TABLE
IRKIND	R ⁴ R _{INC}	INDICATOR FOR THE FOUR LOCATIONS IN A R-K-R INTERVAL
IRUN		RUN IDENTIFICATION
ISC	C	NUMBER OF ELEMENTS

ISCP1	C+1	ISC + 1
ISF	f	DEFINE VIBRATIONAL NONEQUILIBRIUM RANGE
ISFP1	f+1	ISF + 1
ISG	g	DEFINE VIBRATIONAL NONEQUILIBRIUM RANGE
ISGP1	g+1	ISG + 1
ISR	r	NUMBER OF REACTIONS
ISS	s	NUMBER OF SPECIES
ISTART		FINISHED NUMBER OF SETS OF DATA
ISTOP	NSD	NUMBER OF SETS OF DATA
IYST		STARTING STEP INDICATOR
KFIIND(I)		INDICATOR FOR DIRECTION OF REACTION RATE CONSTANT
M	4+g+f	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
M1	M+1	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MSUMJ(J)	m _j	NUMBER OF ELECTRONIC LEVELS, MAXIMUM OF 8
MX	S+g+f	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MX1	MX+1	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MX2	MX+2	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MX3	MX+3	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MX4	MX+4	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
MX5	MX+5	INDEX FOR LOCATING ELEMENTS IN B AND C ARRAYS
NC		INDICATOR FOR FULL OR HALF INTEGRATION STEP SIZE
NOR		NUMBER OF B, C, POLYNOMIALS USED, MAXIMUM OF 3
NUI(I)	ν_i	$\nu_i = \sum_{j=1}^S \nu_{ij}$
NUIJ(I,J)	ν_{ij}	LEFT-HAND SIDE STOICHIOMETRIC COEFFICIENTS
NUIJP(I,J)	ν_{ij}^*	RIGHT-HAND SIDE STOICHIOMETRIC COEFFICIENTS
REND(K)		ENDING VALUE OF A REGION COVERED BY B, C, POLY. A
REST(K)		STARTING VALUE OF A REGION COVERED BY B, C, POLY. A

RHOOP	ρ_0'	REFERENCE DENSITY
SAJ(J)	a_j	$a_j = b_j + \frac{5+2(n_j-1)}{2} \ln T_0'$
SBJ(J)	b_j	CONSTANT FOR CHEMICAL POTENTIAL
SGJL(J,L)	g_{jL}	DEGENERACY OF ELECTRONIC LEVEL L
SHJ(J)	h_j	ENTHALPY INCLUDING HEAT OF FORMATION
SHJO(J)	h_j^0	HEAT OF FORMATION
SHJOP(J)	$h_j^{o'}$	HEAT OF FORMATION
SKBI(I)	k_{B_i}	BACKWARD REACTION RATE CONSTANT
SKFI(I)	k_{F_i}	FORWARD REACTION RATE CONSTANT
SKFIIN(I)	$k_{F_i\infty}$	FORWARD REACTION RATE CONSTANT AT VIBRATIONAL EQUIL.
SPECIK(I,2)		REACTION IDENTIFICATION
SPECJK(J,4)		SPECIES IDENTIFICATION
SUJOOT(J)	μ_j^0/T	CHEMICAL POTENTIAL OF SPECIES
TAUAJ(J)	τ_{aj}	DESCRIBES VIBRATIONAL RELAXATION TIME, TAUJP(J)
TAUBJ(J)	τ_{bj}	DESCRIBES VIBRATIONAL RELAXATION TIME, TAUJP(J)
TAUCJ(J)	τ_{cj}	DESCRIBES VIBRATIONAL RELAXATION TIME, TAUJP(J)
TAUDJ(J)	τ_{dj}	DESCRIBES VIBRATIONAL RELAXATION TIME, TAUJP(J)
TAUJP(J)	τ_j'	VIBRATIONAL RELAXATION TIME OF SPECIES
TEMP	T	TEMPERATURE
THEVJ(J)	θ_{vj}	CHARACTERISTIC VIBRATIONAL TEMPERATURE
THEVJP(J)	θ_{vj}'	CHARACTERISTIC VIBRATIONAL TEMPERATURE
TP(100)		FUNCTION VALUES IN THE TABLE OF B. C.
TSCALE		$CUOP^{**2} * CMWOP / (4.185014E7 * CR0 * CTOP)$
TY(100)		Y VALUES IN THE TABLE OF B. C.
XA		$0.5 * TSCALE$
XB		$6.022E23 / (CR0 * CTOP)$
XC		$CMWOP * CLP / (CUOP * RHOOP)$

XIA2(4)	$\begin{aligned} \text{XIA2}(1) &= U \\ \text{XIA2}(2) &= P \\ \text{XIA2}(3) &= P \\ \text{XIA2}(4) &= A \end{aligned}$	ACCUMULATION COUNTER FOR THE R-K-R INCREMENTAL VALUES
XIA3(J)	$\gamma_j \quad j=1, \dots, s$	ACCUMULATION COUNTER FOR THE R-K-R INCREMENTAL VALUES
XIA4(J)	$\epsilon_j \quad j=f+1, \dots, g$	ACCUMULATION COUNTER FOR THE R-K-R INCREMENTAL VALUES
XLAMJ(J)	λ'_j	VIBRATIONAL RELAXATION DISTANCE
XLCT		$\ln(T)$
XNUI(1)	ν_i	$\nu_i = \sum_{j=1}^s \nu'_{ij}$
XNUIJ(1,J)	ν'_{ij}	LEFT-HAND SIDE STOICHIOMETRIC COEFFICIENTS
XNUIJP(1,J)	ν''_{ij}	RIGHT-HAND SIDE STOICHIOMETRIC COEFFICIENTS
YPREV		LAST VALUE OF y FOR WHICH RESULTS WERE PRINTED
YSTOP	y_{STOP}	YSTOPP/CLP
YSTOPP	y'_{STOP}	VALUE OF STREAMWISE DISTANCE THAT WILL TERMINATE RUN

2. Program Listings

This section presents the complete Fortran IV listings of the main program, as well as all the subroutines of the streamtube program. The functions of all the subroutines are described as follows:

S2 Subroutine START

Performs initialization of variables and definition of constants for the Runge-Kutta-Richardson method.

S3 Subroutine READIN

Reads in and writes out all input data on tape #4.

S4 Subroutine WRITE 1

Writes headings and output formats on tape #4

S5 Subroutine WRITE 2

Writes results on tape #4. If sense switch 6 is on it prints χ_i in each paragraph on tape #4. If sense switch 5 is on it prints Q_{ij} on tape #3.

S6 Subroutine MATRIX

Initializes the matrix for solving simultaneous linear equations.

S7 Subroutine DER

Computes the derivatives of every variable with respect to y .

S8 Subroutine SUB 1

Writes optional outputs on tape #2

S9 Subroutine SUB 2

Computes vibrational relaxation time of the j^{th} species in the $f+1 \rightarrow g$ range.

S10 Subroutine SUB4

Computes the forward rate constant k'_{Fwd}

S11 Subroutine SUB 5

Writes optional outputs on tape #2, if sense switch 1 is on.

S12 Subroutine SUB6

Computes the electron excitation contributions to the thermo-chemical properties $h_j, c_{p,j}, M_j$.

S13 Subroutine TEST 1

Performs output printing step size test, and also makes suitable adjustments on Δy

S14 Subroutine TEST 2

Performs the tests for negative γ_j, ϵ_j, T and A

S15 Subroutine TEST 3

Performs residue test for the Runge-Kutta-Richardson method. If the test fails, it halves the integration step size.

S16 Subroutine OVERFLW

Calls subroutine FPT which is a library subroutine to skip the floating-point overflow trap.

S17 Subroutine LCOUNT

Tape #3 page ejector

S18 Subroutine STOP

Performs computing termination test for each set of data in this run.

S19 Subroutine SIMSOL

Solves the set of simultaneous linear equations by Gauss' method.

S20 Function DERT

Computes derivatives of boundary condition function with respect to streamwise distance, y , from the boundary condition table by Foward Gregory-Newton formula.

S21 Function YINT

Interpolates boundary condition function from the boundary condition table by second-order Newton divided-differences method.

S22 Common and Dimension Statements

Used with the main program and all the subprograms except subroutine SIMSOL, function YINT and function DERT.

ISTART=0	501	1
ASSIGN 5000 TO IGO	501	2
5000 CALL START	501	3
5001 CALL READIN	501	4
IYST=0	501	5
5002 IISFP1=ISF+1	501	6
5003 IISG=ISG	501	7
5004 CALL WRITE 1	501	8
IEXT11=EXTRAJ(11)	501	9
5020 ISCP1=ISC+1	501	10
5030 ISFP1=ISF+1	501	11
5040 ISGP1=ISG+1	501	12
IISF = ISF	501	13
5240 M=4+ISS+ISG-ISF	501	14
5250 M1=M+1	501	15
5260 MX=ISG-ISF+ISS	501	16
5270 MX1=MX+1	501	17
5280 MX2=MX+2	501	18
5290 MX3=MX+3	501	19
5300 MX4=MX+4	501	20
MX5=MX+5	501	21
5301 CALL MATRIX	501	22
5310 DELY=DELYP/CLP	501	23
YSTOP=YSTOPP/CLP	501	24
CR0=1.98647	501	25
5330 TSCALE=(CUOP*CUOP*CMWOP)/(CR0*CTOP*4.185014E+7)	501	26
XC=(CMWOP*CLP)/(CUOP*RHOOP)	501	27
IDELXC=0	501	28
XB=6.022E+23/(CR0*CTOP)	501	29
XA = .5*CUOP*CUOP * CMWOP/(4.185014E+07 *CR0 * CTOP)	501	30
YPREV = -1.0	501	31
5340 DO 5370 J=1,ISS,1	501	32
5350 THEVJ(J)=THEVJP(J)/CTOP	501	33
5360 SHJO(J)=SHJOP(J)/(CR0*CTOP)	501	34
5370 SAJ(J)=SBJ(J)+.5*(5.0+ 2.0*(ETAJ(J)-1.0))*ALOG(CTOP)	501	35
5380 DO 5400 I=1,ISR,1	501	36
5390 DO 5400 J=1,ISS,1	501	37
5400 BETAIJ(I,J) =XNUIJP(I,J)-XNUIJ(I,J)	501	38
5410 DO 5460 I=1,ISR,1	501	39
5420 XNUI(I)=0.0	501	40
5430 BETAI(I)=0.0	501	41
5440 DO 5460 J=1,ISS,1	501	42
5450 XNUI(I)=XNUI(I) +XNUIJ(I,J)	501	43
5460 BETAI(I)=BETAI(I)+BETAIJ(I,J)	501	44
7000 DO 7050 J=1,ISS,1	501	45
7010 DO 7020 L=1,8,1	501	46
7020 CEJLP(J,L)=CEJLPX(J,L)/(3.29820E-24*CTOP)	501	47
7030 EECHJ(J)=0.0	501	48
7040 EECNUJ(J)=0.0	501	49
7050 EECFJ(J)=0.0	501	50
7060 CON1=CR0*CTOP*4.13228E+01	501	51
7070 DO 7120 I=1,ISR,1	501	52
7080 DO 7100 J=1,ISS,1	501	53

7100	IBETIJ(I,J)=BETAIJ(I,J)	501	54
7110	IBETA(I)=BETA(I)	501	55
7120	NUI(I)=XNUI(I)	501	56
5561	A2I(4)=CAOP	501	57
5610	IF (DELY - EXTRAJ(14)) 5611,5611,5619	501	58
5611	WRITE(6,5612) IRUN,AIT(1),DELY	501	59
5612	FORMAT(6X,4HRUN,46.9H SKIPPED,5X6HAT Y =E18.8,5X4HDY =E18.8)	501	60
	ISTART=ISTART+1	501	61
	GO TO 5000	501	62
5619	DO 5670 J=1,3,1	501	63
5660	A1T(J)=A1I(J)	501	64
5661	A1IC(J)=A1I(J)	501	65
	A2IC(J)=A2I(J)	501	66
5670	A2T(J)=A2I(J)	501	67
	A2IC(4)=A2I(4)	501	68
5671	A2T(4)=A2I(4)	501	69
5680	DO 5690 J=1,ISS,1	501	70
5681	A3IC(J)=A3I(J)	501	71
5690	A3T(J)= A3I(J)	501	72
5700	IF(15G-15FP,1) 5613,5710,5710	501	73
5710	DO 5720 J=15FP1,15G,1	501	74
5711	A4IC(J)=A4I(J)	501	75
5720	A4T(J)=A4I(J)	501	76
5613	NC=1	501	77
	IF(1POT.EQ. 0) GO TO 5614	501	78
	DO 5723 J=1,NOR	501	79
	IF(AIT(1).LT.TY(J)) GO TO 5724	501	80
5723	CONTINUE	501	81
5724	IRA1=J	501	82
	IF(IRA1.EQ. 1) IRA1=2	501	83
	IF(IRA1.GE.NOR) IRA1=NOR-1	501	84
	A2T(1BOP)=YINT(TY,TP,IRA1,AIT(1))	501	85
5614	DELYC=DELY/FLOAT(NC)	501	86
5721	DO 6240 IC=1,NC	501	87
5730	DO 5740 J=1,4,1	501	88
5740	XIA2(J)=0.0	501	89
5750	DO 5770 J=1,20,1	501	90
5760	XIA3(J)=0.0	501	91
5770	XIA4(J)=0.0	501	92
5780	DO 6130 III=1,4,1	501	93
5790	IRKIND=III	501	94
5791	IFAIL = 0	501	95
5792	CALL TEST2	501	96
5793	IF(1FAIL) 5610,5800,5810	501	97
5800	CALL DER	501	98
	IF(NC.NE.1 .OR. IC.NE.1) GO TO 5820	501	99
	IFAIL=0	501	100
5810	CALL TEST1	501	101
	IF(1FAIL) 5610,5820,5610	501	102
5820	IF(III.NE. 1) GO TO 5809	501	103
	IF(IC.NE. NC) GO TO 5809	501	104
	DO 5801 J=1,3	501	105
	A1IC(J)=AIT(J)	501	106

5801 A2IC(J)=A2T(J)	S01 107
A2IC(4)=A2T(4)	S01 108
DO 5802 J=1,ISS	S01 109
5802 A3IC(J)=A3T(J)	S01 110
IF((ISS-ISFP1) .LT. 0) GO TO 5809	S01 111
DO 5803 J=ISFP1,ISS	S01 112
5803 A4IC(J)=A4T(J)	S01 113
5809 JJ=0	S01 114
DELYC=DELY/FLOAT(NC)	S01 115
5830 IF((ISS-ISFP1) 5930,5840,5840	S01 116
5840 DO 5920 J=ISFP1,ISS+1	S01 117
5850 JJ=JJ+1	S01 118
5860 TEMP1 =DELYC*B(JJ,M1)	S01 119
5870 XIA4(J)=XIA4(J)+ALPHA(111)*TEMP1	S01 120
5890 A4T(J)=A4IC(J)+BETA(111)*TEMP1	S01 121
5920 CONTINUE	S01 122
5930 TEMP2=0.0	S01 123
DO 6010 J=1,ISS+1	S01 124
5940 JJ=ISS-ISFP1	S01 125
5950 TEMP1=DELYC*B(JJ,M1)	S01 126
5960 XIA3(J)=XIA3(J)+ALPHA(111)*TEMP1	S01 127
5980 A3T(J)=A3IC(J)+BETA(111)*TEMP1	S01 128
6010 TEMP2= TEMP2 + A3T(J)	S01 129
6020 DO 6100 J=1,4+1	S01 130
6030 JJ=MX5-J	S01 131
6040 TEMP1=DELYC*B(JJ,M1)	S01 132
6050 XIA2(J)=XIA2(J)+ALPHA(111)*TEMP1	S01 133
6070 A2T(J)=A2IC(J)+BETA(111)*TEMP1	S01 134
6100 CONTINUE	S01 135
6110 A1T(1)=A1IC(1)+BETA(111)*DELYC	S01 136
6120 A1T(3)=1.0/TEMP2	S01 137
6121 A1T(2)=A2T(3)*A1T(3)*TSSCALE/A2T(2)	S01 138
6130 CONTINUE	S01 139
6140 IF((ISS-ISFP1) 6170,6150,6150	S01 140
6150 DO 6160 J=ISFP1,ISS+1	S01 141
6160 A4T(J)=A4IC(J)+XIA4(J)/6.0	S01 142
6170 TEMP2=0.0	S01 143
6180 DO 6200 J=1,ISS+1	S01 144
6190 A3T(J)=A3IC(J)+XIA3(J)/6.0	S01 145
6200 TEMP2=TEMP2+A3T(J)	S01 146
6210 DO 6220 J=1,4+1	S01 147
6220 A2T(J)=A2IC(J)+XIA2(J)/6.0	S01 148
6230 A1T(3)=1.0/TEMP2	S01 149
A1T(2)=A2T(3)*A1T(3)*TSSCALE/A2T(2)	S01 150
6240 CONTINUE	S01 151
6241 GO TO (6242,6247), NC	S01 152
6242 NC=2	S01 153
6243 DO 6244 J=1,3	S01 154
A1T1(J)=A1T(J)	S01 155
A2T1(J)=A2T(J)	S01 156
A1T(J)=A1T1(J)	S01 157
6244 A2T(J)=A2T1(J)	S01 158
A2T1(4)=A2T(4)	S01 159

A2T(4)=A2T(4)	501 160
DO 6245 J=1,ISS	501 161
A3T1(J)=A3T(J)	501 162
6245 A3T(J)=A3T1(J)	501 163
IF((IISG-IISFP1).LT.0) GO TO 5614	501 164
DO 6246 J=IISFP1,IISG	501 165
A4T1(J)=A4T(J)	501 166
6246 A4T(J)=A4T1(J)	501 167
GO TO 5614	501 168
6247 IF((IISG-IISFP1).LT.0) GO TO 6250	501 169
IFAIL=0	501 170
CALL TEST3(A4T,A4T1,IISFP1,IISG,4)	501 171
IF(IFAIL.NE.0) GO TO 5610	501 172
DO 6248 J=IISFP1,IISG	501 173
6248 A4T(J)=A4T(J)+(A4T(J)-A4T1(J))/15.	501 174
6250 TEMP2=0.0	501 175
IFAIL=0	501 176
CALL TEST3(A3T,A3T1,1,ISS,3)	501 177
IF(IFAIL.NE.0) GO TO 5610	501 178
DO 6249 J=1,ISS	501 179
A3T(J)=A3T(J)+(A3T(J)-A3T1(J))/15.	501 180
TEMP2=TEMP2+A3T(J)	501 181
6249 CONTINUE	501 182
IFAIL=0	501 183
CALL TEST3(A2T,A2T1,1,4,2)	501 184
IF(IFAIL.NE.0) GO TO 5610	501 185
DO 6251 J=1,4	501 186
6251 A2T(J)=A2T(J)+(A2T(J)-A2T1(J))/15.	501 187
A1T(3)=1.0/TEMP2	501 188
A1T(2)=A2T(3)*A1T(3)*TSSCALE/A2T(2)	501 189
6260 GO TO 5613	501 190
END	501 191

```

SUBROUTINE START
CALL CVRFLW
1000 DO 1040 J=1,3,1
1010 A11(J)=0.0
1020 A1T(J)=0.0
1030 A21(J)=0.0
1040 A2T(J)=0.0
1041 A21(4)=0.0
1042 A2T(4)=0.0
1050 DO 1090 J=1,20,1
1060 A31(J)=0.0
1070 A3T(J)=0.0
1080 A41(J)=0.0
1090 A4T(J)=0.0
1100 EOF2=0.0
1110 EOF3=0.0
1120 ALPHA(1)=1.0
1130 ALPHA(2)=2.0
1140 ALPHA(3)=2.0
1150 ALPHA(4)=1.0
1160 BETA(1)=.5
1170 BETA(2)=.5
1180 BETA(3)=1.0
1190 BETA(4)=1.0
1200 RETURN
END

```

```

502 1
502 2
502 3
502 4
502 5
502 6
502 7
502 8
502 9
502 10
502 11
502 12
502 13
502 14
502 15
502 16
502 17
502 18
502 19
502 20
502 21
502 22
502 23
502 24
502 25
502 26

```

	SUBROUTINE READIN	S03	1
	DIMENSION ISGJL(8), NALPIJ(20), IPP(4)	S03	2
	INTEGER QQQ1, QQQ2, QQQ4, QQQ5, QQQ6, QQQ7, PPP1, PPP2	S03	3
C	ALPHAMERIC WORDS IN INTEGER FORM	S03	4
C	QQQ2 = COMP.	S03	5
	QQQ2 = -17508747739	S03	6
C	QQQ4 = NO	S03	7
	QQQ4 = -17997957478	S03	8
C	QQQ5 = YES	S03	9
	QQQ5 = -17997989234	S03	10
C	QQQ6 = FORW	S03	11
	QQQ6 = 24270826544	S03	12
C	QQQ7 = REV	S03	13
	QQQ7 = -10030091312	S03	14
C	IPP(1) = VEL.	S03	15
	IPP(1) = -18074711792	S03	16
C	IPP(2) = RX0	S03	17
	IPP(2) = -17574185264	S03	18
C	IPP(3) = PRESS.	S03	19
	IPP(3) = -8209771673	S03	20
C	IPP(4) = AREA	S03	21
	IPP(4) = -17475916912	S03	22
	112) READ 1130, IRUN, ISS, ISR, ISF, ISG, ISC, IBOP, INDSUM, IDUMP, DELYP, YSTOPP	S03	23
	1130 FORMAT(A6,8I3,2E10,4)	S03	24
	1140 READ 1150, CPOP, RHOOP, CTOP, GAMMA0, CAOP, CUOP, CMXOP, CLP	S03	25
	1150 FORMAT(8E9,5)	S03	26
	1160 READ 1170, (EXTRAJ(J), J=1,15,1)	S03	27
	1170 FORMAT(5E14,7)	S03	28
	1180 DO 1230 J=1,ISS,1	S03	29
	1190 READ 1200, ETAJ(J), SBJ(J), THEVJP(J), SHJOP(J), CNJ(J), CXEJ(J)	S03	30
	1200 FORMAT(6E12,6)	S03	31
	1210 READ 1220, TAUJ(J), TAUBJ(J), TAUCJ(J), TAUDJ(J), (SPECJK(J,K), K=1,4)	S03	32
	1220 FORMAT(4E12,6,4A6)	S03	33
	1230 READ 1240, MSUMJ(J), (SGJL(J,L), L=1,8,1), (CEJ(J,L), L=1,8,1)	S03	34
	1240 FORMAT(12,8F2,0/8E9,5)	S03	35
	1270 DO 1300 I=1,ISR,1	S03	36
	1280 READ 1290, CWI(I), CZI(I), CDI(I), (XNUIJP(I,J), J=1,20),	S03	37
	1 (XNUIJ(I,J), J=1,20), (CAIJ(I,J), J=1,20)	S03	38
	1290 FORMAT(3F2,0,20F1,0,20F1,0,20F1,0)	S03	39
	1300 READ 1310, KFIIND(I), AKFI(I), BKFI(I), CKFI(I), DKFI(I), (SPECIK(I,K),	S03	40
	IK=1,2,1)	S03	41
	1310 FORMAT(14,4E14,7,2A6)	S03	42
	1320 DO 1330 J=1,ISC,1	S03	43
	1330 READ 1340, (ALPIJ(I,J), I=1,ISS,1)	S03	44
	1340 FORMAT(20F2,0)	S03	45
	ISFP1 = ISF + 1	S03	46
	1350 IF (ISG - ISFP1) 1370, 1360, 1360	S03	47
	1360 READ 1170, (A4I(J), J=ISFP1, ISG)	S03	48
	1370 READ 1170, (A3I(J), J=1, ISF)	S03	49
	READ 1170, A1I(1), A1I(2), A2I(3), A2I(2), A2I(1), A1I(3)	S03	50
C	IPOT = 0 - POLY, IPOT = 1 - TABLE	S03	51
	READ 1378, NOK, IPOT	S03	52
	1378 FORMAT(2I3)	S03	53

IF(IPOT,NE, 0) GO TO 1385	S03	84
1079 DO 1382 IR=1,NOR	S03	85
READ 1380, (COP(IR),REST(IR),REND(IR)	S03	86
1380 FORMAT(13,2E14,7)	S03	87
1381 IOOP, IR)=IOOP(IR)+1	S03	88
IOF=IOOP(IR)	S03	89
1382 READ 1170, (COP(IR,J), J=1,IOF)	S03	90
GO TO 2000	S03	91
1385 READ 1386, (TY(IR),TP(IR), IR=1,NOR)	S03	92
1386 FORMAT(2E14,7)	S03	93
2000 WRITE (4,2001)	S03	94
2001 FORMAT(6H1 RUN,5X,1HS,5X,1HR,5X,1HF,5X,1HG,5X,1HC,2X14HBOUNDARY/	S03	95
1OND, 3X,2HEE,8X,5HDELYP,10X,6HYS(OPP,6X,4HDUMP)	S03	96
2008 IF(INDSUM) 2011,2009,2011	S03	97
2009 PPP2=QQQ4	S03	98
2010 GO TO 2012	S03	99
2011 PPP2=QQQ5	S03	100
2012 WRITE (4,2013) (RUN,ISS,ISR,ISF,ISG,ISC,IFP(13OP),PPP2,DELYP,YSTOPPS	S03	101
1, IDUMP	S03	102
2013 FORMAT(1X,A6,15,416,6X,A6,4X,A6,2X,1P2E15,7,4X,11)	S03	103
2014 WRITE (4,2015)	S03	104
2015 FORMAT(1H0,4X,4HCPOP,11, 3HRH0OP,11X,4HCTOP,10X,	S03	105
1 4HCJOP,10X,5HCMWOP,11X,3HCLP)	S03	106
2016 WRITE (4,2017) CPOP,RHOP,CTOP, CUOP,CMWOP,CLP	S03	107
2017 FORMAT(1P5E15,7)	S03	108
2018 WRITE (4,2019) (EXTRAJ(J),J=1, 5, 1)	S03	109
2019 FORMAT(15H0 CONTROLS AND,1P5E15,7)	S03	110
2020 WRITE (4,2021) (EXTRAJ(J),J=6,10,1)	S03	111
2021 FORMAT(15H0 CONSTANTS ,1P5E15,7)	S03	112
WRITE (4,3000) (EXTRAJ(J), J=11,15,1)	S03	113
3000 FORMAT (15X,1P5E15,7)	S03	114
2022 WRITE (4,2023)	S03	115
2023 FORMAT(1H0,11X,1HJ,8X,4HETAJ,11X,3HSBJ,11X,6HTHEVJP,9X,5HSHJOP,11X,5H	S03	116
1,3HCNJ,12X,4HCXOJ)	S03	117
COUNT=11.0	S03	118
2024 DO 2028 J=1,ISS,1	S03	119
WRITE (4,2025) J,ETAJ(J),SBJ(J),THEVJP(J),SHJOP(J),CNJ(J),CXEJ(J)	S03	120
2025 FORMAT(11X,12,2X,1P6E15,7)	S03	121
2026 COUNT=COUNT+1.0	S03	122
2027 CALL LCOUNT	S03	123
2028 CONTINUE	S03	124
2029 WRITE (4,2030)	S03	125
2030 FORMAT(1H0,11X,1HJ,7X,5H,1AUAJ,10X,5HTAUBJ,10X,5HTAUCJ,10X,5HTAUDJ,	S03	126
113X,7H^PECIES)	S03	127
2031 COUNT=COUNT+2.0	S03	128
2032 CALL LCOUNT	S03	129
2033 DO 2038 J=1,ISS,1	S03	130
2034 WRITE (4,2035) J,TAUAJ(J),TAUBJ(J),TAUCJ(J),TAUDJ(J), (SPECJKS	S03	131
1(J,K),K=1,4,1)	S03	132
2035 FORMAT(11X,12,2X,1P4E15,7,4A6)	S03	133
2036 COUNT=COUNT+1.0	S03	134
2037 CALL LCOUNT	S03	135
2038 CONTINUE	S03	136

Line	Code	Statement	Address
2039	WRITE	(4,2040)	503 107
2040	FORMAT	(120H0 J M SGJ1 2 3 4 5 6 7 8CEJ1P 2 3 503 108	
		1 4 5 6 7 8 1503 109	
2041	COUNT=COUNT+2.0		503 110
2042	CALL LCOUNT		503 111
2043	DO 2050 J=1,ISS,1		503 112
2044	DO 2045 L=1,8,1		503 113
2045	ISGJL(L)=SGJL(J,L)		503 114
2046	WRITE (4,2047)J,MSUMJ(J),((ISGJL(L),L=1,8,1),		503 115
		(CEJLPX(J,L),L=1,8,1)	503 116
2047	FORMAT(13,12,2X,8I3,F5.1,1P7E12.5)		503 117
2048	COUNT=COUNT+1.0		503 118
2049	CALL LCOUNT		503 119
2050	CONTINUE		503 120
2059	WRITE (4,2060)		503 121
2060	FORMAT(106H0 1 WI ZI DI NUIJP 1 6 11 16		503 122
		11J 1 5 6 11 16 CAIJ 1 6 11 16)	503 123
2061	COUNT=COUNT+2.0		503 124
2062	DO 2066 I=1,ISR,1		503 125
2063	DO 2066 J=1,ISS,1		503 126
2064	NUIJP(I,J)=XNUIJP(I,J)		503 127
2065	NUIJ(I,J)=XNUIJ(I,J)		503 128
2066	ICAIJ(I,J)=CAIJ(I,J)		503 129
2067	DO 2075 I=1,ISR,1		503 130
2068	NWI=CWI(I)		503 131
2069	NZI=CZI(I)		503 132
2070	NDI=CDI(I)		503 133
2071	WRITE (4,2072)I,NWI,NZI,NDI,(NUIJP(I,J),J=1,20,1),		503 134
		(NUIJ(I,J),J=1,20,1),(ICAIJ(I,J),J=1,20,1)	503 135
2072	FORMAT(3X,12,2X,12,2X,12,2X,12,9X,5I1,1X,5I1,1X,5I1,1X,5I1,9X,5I1,5I1,1X,5I1,1X,5I1,1X,5I1,1X,5I1,1X,5I1,1X,5I1)		503 136
2073	COUNT=COUNT+1.0		503 137
2074	CALL LCOUNT		503 138
2075	CONTINUE		503 139
2076	WRITE (4,2077)		503 140
2077	FORMAT(15H0DIRECTION 1 .6X,4HAKFI,11X,4HBKFI,11X,4HCKFI,11X,14HDKFI,10X,8HREACTION)		503 141
2078	COUNT=COUNT+2.0		503 142
2079	CALL LCOUNT		503 143
2080	DO 2089 I=1,ISR,1		503 144
2081	IF (KFIIND(I))2084,2082,2084		503 145
2082	PPP1=QQQ6		503 146
2083	GO TO 2085		503 147
2084	PPP1=QQQ7		503 148
2085	WRITE (4,2086)PPP1,1, AKFI(1),BKFI(1),CKFI(1),DKFI(1) ,		503 149
		(SPECI	503 150
		IK(I,K),K=1,2,1)	503 151
2086	FORMAT(3X,A6,2X,12,2X,1P4E15,7,3X,2A6)		503 152
2087	COUNT=COUNT+1.0		503 153
2088	CALL LCOUNT		503 154
2089	CONTINUE		503 155
2090	WRITE (4,2091)		503 156
2091	FORMAT(10H0ALPJK J=1,7X,1H5,8X,2H10,8X,2H15,8X,2H20)		503 157
2092	COUNT=COUNT+2.0		503 158

2093 CALL LCOUNT	503 160
2094 DO 2101 J=1,ISC,1	503 161
2095 DO 2096 I=1,ISS,1	503 162
2096 NALPIJ(1)=ALPIJ(1,J)	503 163
2097 WRITE (4,2098)J,(NALPIJ(1),I=1,ISS,1)	503 164
2098 FORMAT(4X,12,2X,2012)	503 165
2099 COUNT=COUNT+1.0	503 166
2100 CALL LCOUNT	503 167
2101 CONTINUE	503 168
A2I(4)=CAOP/CLP**2	503 169
2200 WRITE (4,2210)	503 170
2210 FORMAT(/19H01INITIAL CONDITIONS/1H0,6X1HY,14X1HA,14X1HT,14X1HP,13X	503 171
1 3HRH0,13X1HU,13X2HMW)	503 172
WRITE (4,2017) A1I(1),A2I(4),A1I(2),A2I(3),A2I(2),A2I(1),A1I(3)	503 173
COUNT=COUNT+6.0	503 174
CALL LCOUNT	503 175
WRITE (4,2220) (A3I(J),J=1,ISS)	503 176
2220 FORMAT(1H0,38X2HGX/(1P8E15,7))	503 177
COUNT=COUNT+5.0	503 178
CALL LCOUNT	503 179
IF(1SG-1SFP1) 2250,2230,2230	503 180
2230 WRITE (4,2240) (A4I(J),J=1SFP1,1SG)	503 181
2240 FORMAT(1H0,57X4HEPSJ/(1P8E15,7))	503 182
COUNT=COUNT+5.0	503 183
CALL LCOUNT	503 184
2250 IF(IPOT.NE.0) GO TO 2260	503 185
WRITE (4,2241) IPP(IBOP)	503 186
2241 FORMAT(1H0,15X5HRANGE,32X8HCOP,1 OF ,A6)	503 187
COUNT=COUNT+2.0	503 188
DO 2251 IR=1,NOR	503 189
IOP=100P(IR)	503 190
WRITE (4,2242) REST(IR),REND(IR),(COP(IR,J),J=1,IOP)	503 191
2242 FORMAT(1PE15,7,3H TOE16,7,1H,6E15,7,/(35X,1P6E15,7))	503 192
ELIN=100P(IR)/6+1	503 193
COUNT=COUNT+ELIN	503 194
CALL LCOUNT	503 195
2251 CONTINUE	503 196
GO TO 2262	503 197
2260 WRITE(4,2261) IPP(IBOP), (TY(IR),TP(IR), IR=1,NOR)	503 198
2261 FORMAT(1H0,40X,26HTABLE OF BOUNDARY COND. - ,A6,6H VS. Y, /	503 199
1 (1X2E16,8,1X2E16,8,1X2E16,8,1X2E16,8))	503 200
ELIN=NOR/4+3	503 201
COUNT=COUNT+ELIN	503 202
CALL LCOUNT	503 203
2262 RETURN	503 204
END	503 205

SUBROUTINE WRITE1

DIMENSION G(20)

INTEGER Z1, Z3, Z4, Z5, Z6, Z7, Z8, Z9, BLNK, X1, X2, X3, X4,
1 X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16,
2 X17, X18, X19, X20, G

C Z1 = GAMMA
Z1 = -1757035137
C Z3 = EPS
Z3 = -17997847026
C Z4 = TVJ
Z4 = -17997970783
C Z5 = CHI
Z5 = -17997837849
C Z6 = Q
Z6 = -17997958184
C Z7 = P
Z7 = -17997958183
C Z8 = RHO
Z8 = -17997927974
C Z9 = VEL
Z9 = -17997976752
C BLNK =
BLNK = -17997958192
C X1 TO X20 = 1 TO 20, RESPECTIVELY
X1 = -17209429040
X2 = -17226206256
X3 = -17242983472
X4 = -17259760688
X5 = -17276537904
X6 = -17293315120
X7 = -17310092336
X8 = -17326869552
X9 = -17343646768
X10 = 1086524464
X11 = 1103301680
X12 = 1120078896
X13 = 1136856112
X14 = 1153630328
X15 = 1170410544
X16 = 1187187760
X17 = 1203934976
X18 = 1220742192
X19 = 1237519408
X20 = 2160266288
G(1) = X1
G(2) = X2
G(3) = X3
G(4) = X4
G(5) = X5
G(6) = X6
G(7) = X7
G(8) = X8
G(9) = X9

504	1
504	2
504	3
504	4
504	5
504	6
504	7
504	8
504	9
504	10
504	11
504	12
504	13
504	14
504	15
504	16
504	17
504	18
504	19
504	20
504	21
504	22
504	23
504	24
504	25
504	26
504	27
504	28
504	29
504	30
504	31
504	32
504	33
504	34
504	35
504	36
504	37
504	38
504	39
504	40
504	41
504	42
504	43
504	44
504	45
504	46
504	47
504	48
504	49
504	50
504	51
504	52
504	53

G(10)= X10	SC1	54
G(11)= X11	SC4	55
G(12)= X12	SC4	56
G(13)= X13	SC4	57
G(14)= X14	SC4	58
G(15)= X15	SC1	59
G(16)= X16	SC4	60
G(17)= X17	SC4	61
G(18)= X18	SC4	62
G(19)= X19	SC4	63
G(20)= X20	SC4	64
ISCP1 = ISC+1	SC4	65
COUNT=0.0	SC4	66
100 PRINT 101,IRUN	SC4	67
101 FORMAT(4H1RUN,A6,2X,1HY,13X,4HDELY,9X,26HT RATIO OR TYPE OF FAILURE	SC4	68
1E)	SC4	69
102 WRITE (4,103)IRUN	SC4	70
103 FORMAT(1H1,7X,4HRUN ,A6)	SC4	71
104 WRITE (4,105)	SC4	72
105 FORMAT (1H0,50X,18HFORMAT FOR RESULTS)	SC4	73
1090 WRITE (4,1100)	SC4	74
11000FORMAT(8H Y,14X,1HT,14X,1HA,14X,1HP,13X,3HRHO,13X,1HU,13X,	SC4	75
1 2HMW)	SC4	76
1110 WRITE (4,1120)(Z1,I,I=1,ISS,1)	SC4	77
1120 FORMAT (8(3X,A6,12,4X)/8(3X,A6,12,4X)/4(3X,A6,12,4X))	SC4	78
WRITE(4,10)	SC4	79
10 FORMAT(40X,42HNUMBER DENSITY PER C. C. FOR EACH SPECIES)	SC4	80
CALL SSWTCH(6,K000FX)	SC4	81
GO TO(2000,2002),K000FX	SC4	82
2000 WRITE (4,2001)(Z5,I,I=1,ISR,1)	SC4	83
2001 FORMAT(8(2X,A6,12,5X)/8(2X,A6,12,5X)/8(2X,A6,12,5X)/8(2X,A6,12,5X)	SC4	84
1/8(2X,A6,12,5X))	SC4	85
2002 IF(11SG-11SFP1) 106,1150,1150	SC4	86
1150 WRITE (4,1160)(Z3,I, I=11SFP1,11SG,1)	SC4	87
1160 FORMAT(8(2X,A6,12,5X)/8(2X,A6,12,5X)/4(2X,A6,12,5X))	SC4	88
1170 WRITE (4,1180)(Z4,I, I=11SFP1,11SG,1)	SC4	89
1180 FORMAT (8(2X,A6,12,5X)/8(2X,A6,12,5X)/4(2X,A6,12,5X))	SC4	90
106 WRITE (4,107)	SC4	91
107 FORMAT(/////51X,16HFORMAT FOR DUMPS)	SC4	92
108 WRITE (4,109)	SC4	93
109 FORMAT(1H0,6X,1HY,14X,1HT,14X,2HMW,13X,1HU,13X,3HRHO,13X,1HP,12X,	SC4	94
13HRUN,4X,5HRKIND)	SC4	95
110 WRITE (4,111)(Z1,I,I=1,ISS,1)	SC4	96
111 FORMAT(8(3X,A6,12,4X)/8(3X,A6,12,4X)/4(3X,A6,12,4X))	SC4	97
IF(11SG-11SFP1)130,112,112	SC4	98
112 WRITE (4,113)(Z3,I, I=11SFP1,11SG,1)	SC4	99
113 FORMAT(8(2X,A6,12,5X)/8(2X,A6,12,3X)/4(2X,A6,12,5X))	SC4	100
130 DO 114 J=1,ISS,1	SC4	101
114 WRITE (4,115)(J,K=1,8,1)	SC4	102
115 FORMAT(10H SUJOOT,12,8X,4HCCPJ,12,9X,4HEPSJ,12,9X,3NSHJ,12,9X,6S	SC4	103
1HEPSJIN,12,8X,4HCTVJ,12,8X,3HXLAMJ,12,9X,3HCVJ,12)	SC4	104
116 DO 117 I=1,ISR,1	SC4	105
117 WRITE (4,118)(I,K=1,7,1)	SC4	106

118	FORMAT(3X,7HDELT F1,12,7X,4HCKP1,12,10X,3HCK1,12,9X,6HCKF11N,12,	504	107
	1 7X,4HCKF1,12,9X,4HCKB1,12,9X,4HCK11,12)	504	108
119	DO 120 I=1,ISS,1	504	109
120	WRITE (4,121)(Z6,I,J, J='SCP1,ISS,1)	504	110
121	FORMAT (8(3Y,A6,12,12,2X))	504	111
122	WRITE (4,123)	504	112
123	FORMAT(120H (PARTIALS OF VARIABLES LISTED BELOW WRT Y	504	113
	1 B/LINE	504	114
124	IF(11SG-11SFP1)127,125,125	504	115
125	WRITE (4,126)(Z3,G(I),I=11SFP1,11SG,1),(Z1,G(I),I=1,ISS,1),Z7,BLANK	504	116
	1K,Z8,BLANK,Z9,BLANK	504	117
126	FORMAT (8(3X,A6,A6))	504	118
	GO TO 128	504	119
127	WRITE (4,126)(Z1,G(I),I=1,ISS,1),Z7,BLANK,Z8,BLANK,Z9,BLANK	504	120
128	WRITE (4,103)IRUN	504	121
	RETURN	504	122
	END	504	123

SUBROUTINE WRITE 2	S05	1
DIMENSION DENN(20)	S05	2
1007 WRITE (4,1008)A11(1),A11(2),A21(4),A21(3),A21(2),A21(1),A11(3)	S05	3
1008 FORMAT (///1P7E15.6)	S05	4
1009 WRITE (4,1010) (A31(J),J=1,ISS,1)	S05	5
1010 FORMAT (1P8E15.6)	S05	6
DO 10 J=1,ISS	S05	7
10 DENN(J)=6.0230E23*A31(J)*A21(2)*RH00P/CMW0P	S05	8
WRITE(4,1010) (DENN(J),J=1,ISS)	S05	9
CALL SSWTCH(6,K000FX)	S05	10
GO TO(3000,3001),K000FX	S05	11
3000 WRITE (4,1010) (CHI1(I),I=1,ISR,1)	S05	12
3001 IF (IISG-IISFP1) 3004,3002,3002	S05	13
3002 WRITE (4,1010) (A41(J), J=IISFP1,IISG,1)	S05	14
3003 WRITE (4,1010) (CTVJ(J), J=IISFP1,IISG,1)	S05	15
3004 CALL SSWTCH(5,K000FX)	S05	16
GO TO(3005,3010),K000FX	S05	17
3005 WRITE (3,3006)A11(1),IRUN	S05	18
3006 FORMAT(///6X,1PE15.6,9X,A6)	S05	19
3007 C0 3008 II=1,ISR,1	S05	20
3008 WRITE (3,1010) (CQIJ(II,J),J=1SCP1,ISS,1)	S05	21
3009 EOF3=1.0	S05	22
3010 COUNT=COUNT+1.0	S05	23
3011 IF (COUNT-EXTRAJ(1)) 3015,3012,3012	S05	24
3012 WRITE (4,3013)IRUN	S05	25
3013 FORMAT(12H1 RUN :A6)	S05	26
3014 COUNT=0.0	S05	27
3015 CONTINUE	S05	28
3016 CALL STOP	S05	29
YPREV = A11(1)	S05	30
3019 RETURN	S05	31
END	S05	32

```

      SUBROUTINE MATRIX
1000 DO 1002      L=1+N.1
1001 DO 1002      LL=1+M1.1
1002 C(L,LL)=0.0
1003 MM=1
1004 IF (IISG-IISFP1)1008,1005,1005
1005 DO 1007      J=IISFP1,IISG.1
1006 C(MM,MM)=1.0
1007 MM=MM+1
1008 DO 1012      J=1,ISC.1
1009 DO 1011      JJ=1,ISS.1
1010 JJJ1=IISG-IISF+JJ
1011 C(MM,JJJ1)=ALPIJ(JJ,J)
1012 MM=MM+1
1013 DO 1015      J=ISCP1,ISS.1
1014 C(MM,MM)=1.0
1015 MM=MM+1
      JP=MX5-IBOP
      C(MX1,JP)=1.0
1017 C(MX2,MX2)=1.0
1018 C(MX3,MX3)=1.0
1019 RETURN
      END

```

```

506 1
506 2
506 3
506 4
506 5
506 6
506 7
506 8
506 9
506 10
506 11
506 12
506 13
506 14
506 15
506 16
506 17
506 18
506 19
506 20
506 21
506 22
506 23

```

SUBROUTINE DER		S07A	1
C	NONPREFERENTIAL MODEL FOR STREAM TUBE	S07A	2
1000	CALL SUB1	S07A	3
1001	IF (INDSUM) 1003, 1003, 1002	S07A	4
1002	CALL SUB3	S07A	5
1003	TEMP=A1T(2)	S07A	6
1010	XLCT=ALOG(TEMP)	S07A	7
1020	CTP=CTOP*TEMP	S07A	8
1030	CPP = A2T(3)*CUOP*CUOP*RHOO	S07A	9
1040	IF (ISF) 1200, 1200, 1050	S07A	10
1050	DO 1190 J=1, ISF, 1	S07A	11
1060	IF (ETAJ(J)-1.0) 1120, 1070, 1120	S07A	12
1070	SUJ00T(J)=-SAJ(J)-2.5*XLCT+SHJ0(J)/TEMP-EECNUJ(J)	S07A	13
1080	CCPJ(J)=2.5+EECCPJ(J)	S07A	14
1090	EPSJ(J)=0.0	S07A	15
1100	SHJ(J)=2.5*TEMP+SHJ0(J)+EECHJ(J)	S07A	16
1110	GO TO 1190	S07A	17
1120	AA=EXP(THEVJ(J)/TEMP)	S07A	18
1130	AA1=.5*(5.0+2.0 * (ETAJ(J)-1.0))	S07A	19
1140	AA2= ETAJ(J)-1.0	S07A	20
1150	SUJ00T(J)=-SAJ(J)-AA1*XLCT+AA2*ALOG(1.0-1.0/AA)+SHJ0(J)/TEMP-	S07A	21
1	EECNUJ(J)	S07A	22
1160	EPSJ(J)=THEVJ(J)/(AA-1.0)	S07A	23
1170	CCPJ(J)=AA1+AA2*EPSJ(J)**2*AA/TEMP**2 +EECCPJ(J)	S07A	24
1180	SHJ(J)=AA1*TEMP+AA2*EPSJ(J)+SHJ0(J)+EECHJ(J)	S07A	25
1190	CONTINUE	S07A	26
1200	IF (ISS-ISGP1) 1360, 1210, 1210	S07A	27
1210	DO 1350 J=ISGP1, ISS, 1	S07A	28
1220	IF (ETAJ(J)-1.0) 1280, 1230, 1280	S07A	29
1230	SUJ00T(J)=-SAJ(J)-2.5*XLCT+SHJ0(J)/TEMP-EECNUJ(J)	S07A	30
1240	CCPJ(J)=2.5+EECCPJ(J)	S07A	31
1250	EPSJ(J)=0.0	S07A	32
1260	SHJ(J)=2.5*TEMP+SHJ0(J)+EECHJ(J)	S07A	33
1270	GO TO 1350	S07A	34
1280	AA=EXP(THEVJ(J)/TEMP)	S07A	35
1290	AA1=.5 * (5.0 +2.0 * (ETAJ(J)-1.0))	S07A	36
1300	AA2=ETAJ(J)-1.0	S07A	37
1310	SUJ00T(J)=-SAJ(J)-AA1*XLCT+AA2*ALOG(1.0-1.0/AA)+SHJ0(J)/TEMP-	S07A	38
1	EECNUJ(J)	S07A	39
1320	EPSJ(J)=THEVJ(J) / (AA-1.0)	S07A	40
1330	CCPJ(J)=AA1+AA2*EPSJ(J)**2*AA/TEMP**2 +EECCPJ(J)	S07A	41
1340	SHJ(J)=AA1*TEMP +AA2*EPSJ(J) +SHJ0(J)+EECHJ(J)	S07A	42
1350	CONTINUE	S07A	43
1360	IF (ISG-ISFP1) 1560, 1361, 1361	S07A	44
1361	CALL SUB2	S07A	45
1370	DO 1540 J= ISFP1, ISG, 1	S07A	46
1380	BB=EXP(THEVJ(J)/TEMP)	S07A	47
	BB1= ETAJ(J)-1.0	S07A	48
	BB2=.5*(5.0+2.0*BB1)	S07A	49
1390	EPSJIN(J)=THEVJ(J)/(BB-1.0)	S07A	50
1400	CTVJ(J)=THEVJ(J)/ALOG((THEVJ(J)+A4T(J))/A4T(J))	S07A	51
1410	CCPJ(J)=BB2 +EECCPJ(J)	S07A	52
1420	SUJ00T(J)=-SAJ(J)-BB2*XLCT+BB1*ALOG(1.0-1.0/BB)+SHJ0(J)/TEMP-	S07A	53

1	1430	CHJ(J)=THEVJ(J)*(1.0/CTVJ(J)-1.0/TEMP)	S07A 54
	1440	SHJ(J)=BB2*TEMP+BB1*A3T(J)+SHJO(J)+EECHJ(J)	S07A 55
	1460	XLAMJ(J)=TAUJP(J)*CUOP*A2T(1)/CLP	S07A 56
	1470	IF(CWJ(J))1520,1471,1520	S07A 57
	1471	CVJ(J)=1.0	S07A 58
	1472	GO TO 1540	S07A 59
	1520	CVJ(J) = ((1.0-EXP(-CNJ(J)*CWJ(J)))*(EXP(THEVJ(J)/CTVJ(J))-1.0))	S07A 60
	1	/ (CNJ(J)*(EXP(CWJ(J))-1.0)*(BB-1.0))	S07A 61
	1540	CONTINUE	S07A 62
	1560	CALL SUB4	S07A 63
	1561	DO 1790 II=1,ISR,1	S07A 64
	1570	DF10(II)=0.0	S07A 65
	1580	DO 1590 J=1,ISS,1	S07A 66
	1590	DF10(II)=DF10(II)+BETA1J(II,J)*SUJ00T(J)	S07A 67
	1600	CKP1(II)=EXP(-DF10(II))	S07A 68
	1610	CKI(II)=CKP1(II)/((TEMP*CON1)**IBETA1(II))	S07A 69
	1630	TEMP1=1.0	S07A 70
	1640	IF(1SG-1SFP1) 1670,1650,1650	S07A 71
	1650	DO 1660 J=1SFP1,1SG,1	S07A 72
	1660	TEMP1=TEMP1*CVJ(J)**1CA1J(II,J)	S07A 73
	1670	SKFI(II)=SKFIIN(II) *TEMP1	S07A 74
	1680	SKBI(II) = SKFIIN(II) / CKI(II)	S07A 75
		XXTEMP = TEMP1	S07A 76
	1690	TEMP1=1.0	S07A 77
	1700	DO 1711 J=1,ISS,1	S07A 78
	1701	IF (IBET1J(II,J))1702,1703,1702	S07A 79
	1702	IF(A3T(J))1710,1719,1710	S07A 80
	1703	IF(XNU1J(II,J))1702,1711,1702	S07A 81
	1710	TEMP1=TEMP1* (A3T(J)/CHWOP)**IBET1J(II,J)	S07A 82
	1711	CONTINUE	S07A 83
		GOTO1720	S07A 84
	1719	TEMP1=0.0	S07A 85
	1720	CH11(II)=1.0-((RH00P *A2T(2))**IBETA1(II))*TEMP1 / (CKI(II)*XXTEMP	S07A 86
	1)		S07A 87
C		EXTRAJ(4) - UPPER BOUND, EXTRAJ(5) - LOWER BOUND OF CH11	S07A 88
		IF(CH11T(II) .EQ. 0.0) GO TO 1725	S07A 89
		IF(ABS(CH11(II)).LT.EXTRAJ(5)) CH11(II)=0.0	S07A 90
		GO TO 1726	S07A 91
	1725	IF(ABS(CH11(II)).LT. EXTRAJ(4)) CH11(II)=0.0	S07A 92
	1726	CONTINUE	S07A 93
		TEMP2=0.0	S07A 94
		IF (CZ1(II))1732,1732,1730	S07A 95
	1730	DO 1731 J=1,ISS,1	S07A 96
	1731	TEMP2=TEMP2+(BETA1J(II,J)+1.0)*XNU1J(II,J)*A3T(J)	S07A 97
	1732	TEMP1=0.0	S07A 98
		IF (CW1(II))1733,1733,1734	S07A 99
	1733	IF(CD1(II))1740,1740,1734	S07A100
	1734	TEMP1=1.0	S07A101
		DO 1737 J=1,ISS,1	S07A102
		IF (A3T(J))1736,1735,1736	S07A103
	1735	IF (NU1J(II,J))1736,1737,1736	S07A104
	1736	TEMP1=TEMP1 * (A3T(J)/CHWOP)**NU1J(II,J)	S07A105
			S07A106

1737	CONTINUE	S07A107
	TEMP1= C _{YI} (11)*TEMP1*(RHOOP*A2T(2))/**NUJ(11) + C _{DI} (11)*TEMP1*	S07A108
1	(RHOJP*A2T(2))*2/(A1T(3)*CMWOP)	S07A109
1740	TEMP2= C _{ZI} (11)*TEMP2*(RHOOP*A2T(2))*2 /CMWOP**2	S07A110
1738	DO 1739 J=ISCP1,ISS,1	S07A111
1739	C _{QIJ} (11,J)=XC*BETA1J(11,J)*SKFI(11)*CHI(11)*(TEMP1 +TEMP2*A2T(J))	S07A112
1790	CONTINUE	S07A113
1800	DO 1820 L=1,M,1	S07A114
1810	DO 1820 LL=1,M1,1	S07A115
1820	B(L,LL)= C(L,LL)	S07A116
	B(MX1,M1)=0.0	S07A117
	IF(IPOT,NE,0) GO TO 1870	S07A118
	IRA=1	S07A119
	GO TO (1828,1827,1825), NOR	S07A120
1826	IF(A1T(1),GT,REND(2)) IRA=IRA+1	S07A121
1827	IF(A1T(1),GT,REND(1)) IRA=IRA+1	S07A122
1828	MN=100P(IRA)-1	S07A123
	IF(MN) 1830,1830,1823	S07A124
1823	DO 1829 J=1,MN	S07A125
	MN1=100P(IRA)-J+1	S07A126
	FMN=MN1-1	S07A127
1829	B(MX1,M1)=B(MX1,M1) *A1 ⁻ (1)+FMN*COP(IRA,MN1)	S07A128
	GO TO 96	S07A129
1870	DO 1871 J=1,NOR	S07A130
	IF(A1T(1),LT,TY(J)) GO TO 1872	S07A131
1871	CONTINUE	S07A132
1872	IRA=J	S07A133
	IF(IRA,EG,1) IRA=2	S07A134
	IF(IRA,GE,NOR) IRA=NOR-1	S07A135
	B(MX1,M1)=DERT(TY,TP,IRA,A1T(1))	S07A136
96	CALL SSWTCH(4,KSW)	S07A137
	GO TO (97,1830),KSW	S07A138
97	WRITE(6,98) A1T(1),IRA,B(MX1,M1),DELY	S07A139
98	FORMAT(4H Y =E16.8,6X,5HIRA =14.6X4HDF =E16.8,6X,4HGY =E16.0)	S07A140
1630	B(MX2,MX4)= A2T(2)*A2T(1)	S07A141
1840	NN=1	S07A142
1350	IF (15G-15FP1) 1890,1860,1850	S07A143
1860	DO 1880 J=115FP1,115G,1	S07A144
	TEMP1=0.0	S07A145
	TEMP2=0.0	S07A146
	DO 1861 II=1,ISR,1	S07A147
	TEMP3=(C _{AIJ} (11,J)*C _{QIJ} (11,J))/(A3T(J)*A2T(2)*A2T(1)*CHI(11))	S07A148
	TEMP1=TEMP1+TEMP3	S07A149
1861	TEMP2=TEMP2+TEMP3*(1.0-CHI(11))	S07A150
	TEMP3=THEVJ(J)*(1.0/CTVJ(J)-1.0/TEMP)	S07A151
	TEMP4=(EPSJIN(J)-A4T(J))/XLAMJ(J)	S07A152
	TEMP5=(THEVJ(J)/(EXP(TEMP3)-1.0) -(CNJ(J)*THEVJ(J)) /	S07A153
1	(EXP(CNJ(J)*TEMP3)-1.0) - A4T(J)) *TEMP1	S07A154
	TEMP2 = -(0.5*(CNJ(J)-1.0)*THEVJ(J)-A4T(J))* TEMP2	S07A155
	B(MN,M1) = TEMP4+TEMP5+TEMP2	S07A156
1880	NN=NN+1	S07A157
1890	DO 1940 J=ISCP1,ISS,1	S07A158
1900	JJ=115G-115F+J	S07A159

```

1910 TEMP1=0.0
1920 DO 1930 I=1,ISR,1
1930 TEM=1+TEMP1+COIJ(1,J)
1940 B(JJ,M1)=TEM/(A2T(2)*A2T(1))
1950 TEMP1=0.0
1960 DO 1970 J=1,ISS,1
1970 TEMP1=TEMP1+A3T(J)*CCPJ(J)
1980 TEMP1=TEMP1*TSCALE
1990 P(MX4,MX2)=TEMP1*A1T(3)/A2T(2)
2000 B(MX4,MX3)=-(TEMP1*A1T(3)*A2T(3))/A2T(2)+2
2010 B(MX4,MX4)=TSCALE*A2T(1)
2020 LLL=0
2030 IF(11SG-11SFP1)2070,2040,2040
2040 DO 2060 J=11SFP1,11SG,1
2050 LLL=LLL+1
2060 B(MX4,LLL)=A3T(J)*(EYAJ(J)-1.0)
2070 XISS=ISS
2080 DO 2100 J=1,ISS,1
2090 KKK=11SG-11SF+J
2100 B(MX4,KKK)=SHJ(J)-(TEMP1*A1T(3)**2*A2T(3))/A2T(2)
2101 B(MX3,MX4)=A2T(2)/A2T(1)
      B(M3,MX1)=A2T(2)/A2T(4)
      CALL SSWTCH(3,NSW3)
      IF(NSW3,EQ,1) PRINT 2105,A1T(1),DELY
2105 FORMAT(7HOAT Y *E12.8, 3X4HDY *E18.8)
10 CALL SIMSOL ( B,M,45)
10 CALL SUB5
30 RETURN
END

```

507A160
507A161
507A162
507A163
507A164
507A165
507A166
507A167
507A168
507A169
507A170
507A171
507A172
507A173
507A174
507A175
507A176
507A177
507A178
507A179
507A180
507A181
507A182
507A183
507A184
507A185
507A186
507A187
507A188

C	SUBROUTINE DER	5075	1
	PREFERENTIAL MODEL FOR STREAM TUBE	5076	2
	IF (YPREV) 6000, 1000, 1000	5077	3
6000	IF (ISG-1SFPI) 1000, 6001, 6001	5078	4
6001	WRITE (4, 6007)	5079	5
6007	FORMAT (5X, 1HN, 9X, 1HU, 11X, 2HWE, 11X, 4HWEXE, 10X, 4HWEYE, 10X, 4HWEZE)	5076	6
	DO 5004 J=1SFPI, ISG, 1	5078	7
	READ 6002, NTEMP(J), UTEMP(J), WE(J), WEXE(J), WEYE(J), WEZE(J)	5078	8
6002	FORMAT (12, 5E14, 7)	5078	9
	WRITE (4, 6006) NTEMP(J), UTEMP(J), WE(J), WEXE(J), WEYE(J), WEZE(J)	5078	10
6006	FORMAT (16, 1P5E14, 7)	5075	11
	NTEMP(J)=NTEMP(J)+1	5078	12
	TEMP2=0.0	5078	13
	TEMP3=0.0	5078	14
	KKP = NTEMP(J)	5078	15
	DO 6003 KK=1, KKP, 1	5078	16
	XKK = KK	5078	17
	EVJP(KK, J)=(XKK-.5)*(WE(J)+(XKK-.5)*(-WEXE(J)+(XKK-.5)*(WEYE(J)	5078	18
1	+WEZE(J)*(XKK-.5))))	5078	19
	IF (KK-1) 6009, 6008, 6009	5078	20
6008	TEMP20=EVJP(1, J)	5078	21
6009	EVJP(KK, J) = EVJP(KK, J)-TEMP20	5078	22
	TEMP4=EXP(EVJP(KK, J)/UTEMP(J))	5078	23
	TEMP2=TEMP2+EVJP(KK, J)*TEMP4	5078	24
6003	TEMP3=TEMP3+TEMP4	5078	25
	QJBAR(J)=(1.987322 *TEMP2)/(TEMP3*CTOP*CRO)	5078	26
6004	QJMUJ(J)=TEMP3	5078	27
1000	CALL SUB1	5078	28
1001	IF (INDSUM) 1003, 1003, 1002	5078	29
1002	CALL SUB6	5078	30
1003	TEMP=A1T(2)	5078	31
1010	XLCT=ALOG(TEMP)	5078	32
1020	CTP=CTOP*TEMP	5078	33
1030	CPP = A2T(3)*CUOP+CUOP*RHOOP	5078	34
1040	IF (ISF) 1200, 1200, 1050	5078	35
1050	DO 1190 J=1, ISF, 1	5078	36
1060	IF (ETAJ(J)-1.0) 1120, 1070, 1120	5078	37
1070	SUJCOT(J)=-SAJ(J)-2.5*XLCT+SHJO(J)/TEMP-EECNUJ(J)	5078	38
1080	CCPJ(J)=2.5+EECCPJ(J)	5078	39
1090	EPSJ(J)=0.0	5078	40
1100	SHJ(J)=2.5*TEMP+SHJO(J)+EECHJ(J)	5078	41
1110	GO TO 1190	5078	42
1120	AA=EXP(THVJ(J)/TEMP)	5078	43
1130	AA1=.5*(5.0+2.0 * (ETAJ(J)-1.0))	5078	44
1140	AA2= ETAJ(J), -1.0	5078	45
1150	SUJCOT(J)=-SAJ(J)-AA1*XLCT+AA2*ALOG(1.0-1.0/AA)+SHJO(J)/TEMP-	5078	46
1	EECNUJ(J)	5078	47
1160	EPSJ(J)=THVJ(J)/(AA-1.0)	5078	48
1170	CCPJ(J)=AA1+AA2*EPSJ(J)**2*AA/TEMP**2 +EECCPJ(J)	5078	49
1180	SHJ(J)=AA1*TEMP+AA2*EPSJ(J)+SHJO(J)+EECHJ(J)	5078	50
1190	CONTINUE	5078	51
1200	IF (ISS-1SGPI) 1360, 1210, 1210	5078	52
1210	DO 1350 J=1SGPI, ISS, 1	5078	53

1220	IP(ETAJ(J)-1.0) 1280,1230,1290	5078	54
1230	SUJ00T(J)=-SAJ(J)-2.5*XLCT+SHJ0(J)/TEMP-EECNUJ(J)	5078	55
1240	CCPJ(J)=2.5+EECCPJ(J)	5078	56
1250	EPSJ(J)=0.0	5078	57
1260	SHJ(J)=2.5*TEMP+SHJ0(J)+EECHJ(J)	5078	58
1270	GO TO 1350	5078	59
1280	AA=EXP(THEVJ(J)/TEMP)	5078	60
1290	AA1= .5 * (5.0 +2.0 * (ETAJ(J)-1.0))	5078	61
1300	AA2=ETAJ(J)-1.0	5078	62
1310	SUJ00T(J)=-SAJ(J)-AA1*XLCT+AA2*ALOG(1.0-1.0/AA)+SHJ0(J)/TEMP-	5078	63
1	EECNUJ(J)	5078	64
1320	EPSJ(J)=THEVJ(J) / (AA-1.0)	5078	65
1330	CCR(J)=AA1+AA2*EPSJ(J)**2*AA/TEMP**2 +EECCPJ(J)	5078	66
1340	SHJ(J)=AA1*TEMP +AA2*EPSJ(J) +SHJ0(J)+EECHJ(J)	5078	67
1350	CONTINUE	5078	68
1360	IF (ISG-1SFP1) 1560,1361,1361	5078	69
1361	CALL SUB2	5078	70
1370	DO 1540 J= 1SFP1,1SG,1	5078	71
1380	BB=EXP(THEVJ(J)/TEMP)	5078	72
	BB1= ETAJ(J)-1.0	5078	73
	BB2= .5*(5.0+2.0*BB1)	5078	74
1390	EPSJIN(J)=THEVJ(J)/(BB-1.0)	5078	75
1400	CTVJ(J)=THEVJ(J)/ALOG((THEVJ(J)+A4T(J))/A4T(J))	5078	76
1410	CCPJ(J)=BB2 +EECCPJ(J)	5078	77
1420	SUJ00T(J)=-SAJ(J)-BB2*XLCT+BB1*ALOG(1.0-1.0/BB)+SHJ0(J)/TEMP-	5078	78
1	EECNUJ(J)	5078	79
1430	CVJ(J)=THEVJ(J)*(1.0/CTVJ(J)-1.0/TEMP)	5078	80
1440	SHJ(J)=BB2*TEMP+BB1*A4T(J)+SHJ0(J)+EECHJ(J)	5078	81
1460	XLAMJ(J)=TAUJP(J)*CUOP*A2T(1)/CLP	5078	82
	TFJP(J)=1.0/(1.0/(CTVJ(J)*CTOP)-1.0/CTP-1.0/UTEMP(J))	5078	83
	TEMP2=0.0	5078	84
	TEMP3=0.0	5078	85
	TEMP4=0.0	5078	86
	TEMP5=0.0	5078	87
	KKP=NTEMP(J)	5078	88
DO 6005	KK=1,KKP,1	5078	89
	TEMP6=EXP(-EVJP(KK,J)/TFJP(J))	5078	90
	TEMP7=EXP(-EVJP(KK,J) / CTP)	5078	91
	TEMP8=EXP(-EVJP(KK,J)/ (CTVJ(J)*CTOP))	5078	92
	TEMP2=TEMP2+EVJP(KK,J)*TEMP6	5078	93
	TEMP3=TEMP3+ TEMP6	5078	94
	TEMP4= TEMP4 + TEMP7	5078	95
6005	TEMP5=TEMP5 + TEMP8	5078	96
	EBARJ(J)=(1.987322*TEMP2) / (TEMP3*CR0*CTOP)	5078	97
1470	IF(CVJ(J))1520,1471,1520	5078	98
1471	CVJ(J)=1.0	5078	99
1472	GO TO 1540	5078	100
1520	CVJ(J)=(TEMP4*TEMP3)/(TEMP5* QJMUJ(J))	5078	101
1540	CONTINUE	5078	102
1560	CALL SUB4	5078	103
1561	DO 1790 II=1,1SR,1	5078	104
1570	DFJO(II)=0.0	5078	105
1590	DO 1590 J=1,1SS,1	5078	106

1590	DF10(11)=DF10(11)+BETA1J(11,J)*SUJ00T(J)	S078107
1600	CKP1(11)=EXP(-DF10(11))	S078108
1610	CK1(11)=CKP1(11)/((TEMP*CON1)**IBETA1(11))	S078109
1630	TEMP1=1.0	S078110
1640	IF(17G-15FP1) 1670,1650,1650	S078111
1650	DO 1660 J=15FP1,15G,1	S078112
1660	TEMP1=TEMP1*CVJ(J)**1CA1J(11,J)	S078113
1670	SKF1(11)=SKFI1N(11) *TEMP1	S078114
1680	SKB1(11) = SKF11N(11) / CK1(11)	S078115
	XXTEMP = TEMP1	S078116
1690	TEMP1=1.0	S078117
1700	DO 1711 J=1,ISS,1	S078118
1701	IF (IBETA1J(11,J))1702,1703,1702	S078119
1702	IF(A3T(J))1710,1719,1710	S078120
1703	IF(XNUIJ(11,J))1702,1711,1702	S078121
1710	TEMP1=TEMP1* (A3T(J)/CMWOP)**IBETA1J(11,J)	S078122
1711	CONTINUE	S078123
	GOTO1720	S078124
1719	TEMP1=0.0	S078125
1720	CH11(11)=1.0-((RHOOP *A2T(2))**IBETA1(11))*TEMP1 / (CK1(11)*XXTEMP	S078126
	1)	S078127
C	EXTRAJ(4) - UPPER BOUND, EXTRAJ(5) - LOWER BOUND OF CH11	S078128
	IF(CH11T(11) .EQ. 0.0) GO TO 1725	S078129
	IF(ABS(CH11(11)).LT.EXTRAJ(5)) CH11(11)=0.0	S078130
	GO TO 1726	S078131
1725	IF(ABS(CH11(11)).LT. EXTRAJ(4)) CH11(11)=0.0	S078132
1726	CONTINUE	S078133
	TEMP2=0.0	S078134
	IF (CZ1(11))1732,1732,1730	S078135
1730	DO 1731 J=1,ISS,1	S078136
1731	TEMP2=TEMP2+(BETA1J(11,J)+1.0)*XNUIJ(11,J)*A3T(J)	S078137
1732	TEMP1=0.0	S078138
	IF (CWI(11))1733,1733,1734	S078139
1733	IF(CD1(11))1740,1740,1734	S078140
1734	TEMP1=1.0	S078141
	DO 1737 J=1,ISS,1	S078142
	IF (A3T(J))1736,1735,1736	S078143
1735	IF (NUIJ(11,J))1736,1737,1736	S078144
1736	TEMP1=TEMP1 * (A3T(J)/CMWOP)**NUIJ(11,J)	S078145
1737	CONTINUE	S078146
	TEMP1= CWI(11)*TEMP1*(RHOOP*A2T(2))**NUI(11) + CD1(11)*TEMP1*	S078147
	1 (RHOOP*A2T(2))**2/(AIT(3)*CMWOP)	S078148
1740	TEMP2= CZ1(11)*TEMP2*(RHOOP*A2T(2))**2 /CMWOP *2	S078149
1738	DO 1739 J=1SCP1,ISS,1	S078150
1739	CQ1J(11,J)=XC*BETA1J(11,J)*SKFI(11)*CH11(11)*(TEMP1 +TEMP2*A3T(J))	S078151
1790	CONTINUE	S078152
1800	DO 1820 L=1,M,1	S078153
1810	DO 1820 LL=1,M1,1	S078154
1820	B(L,LL)= C(L,LL)	S078155
	B(MX1,M1)=0.0	S078156
	IF(IPOT.NE.0) GO TO 1870	S078157
	IRA=1	S078158
	GO TO (1828,1827,1826), NOR	S078159

1826 IF(A1T(1).GT.REND(2)) IRA=IRA+1	5078160
1827 IF(A1T(1).GT.REND(1)) IRA=IRA+1	5078161
1828 MN=100P(IRA)-1	5078162
IF(MN) 1830,1830,1823	5078163
1823 DO 1829 J=1,MN	5078164
MN1=100P(IRA)-J+1	5078165
FMN=MN1-1	5078166
1829 B(MX1,M1)=B(MX1,M1)*A1T(1)+FMN#COP(IRA,MN1)	5078167
GO TO 96	5078168
1870 DO 1871 J=1,NOR	5078169
IF(A1T(1).LT.TY(J)) GO TO 1872	5078170
1871 CONTINUE	5078171
1872 IRA=J	5078172
IF(IRA.EQ.1) IRA=2	5078173
IF(IRA.GE.NOR) IRA=NOR-1	5078174
B(MX1,M1)=DERT(TY,TP,IRA,A1T(1))	5078175
96 CALL SSWTCH(4,KSW)	5078176
GO TO (97,1830),KSW	5078177
97 WRITE(6,98) A1T(1),IRA,B(MX1,M1),DELY	5078178
98 FORMAT(4H Y =E16.8,6X,5H IRA =14,6X,4H DY =E16.8)	5078179
1830 B(MX2,MX4)=A2T(2)*A2T(1)	5078180
1840 NN=1	5078181
1850 IF (11SG-11SFP1) 1890,1860,1860	5078182
1860 DO 1880 J=11SFP1,11SG,1	5078183
TEMP1=0.0	5078184
TEMP2=0.0	5078185
DO 1861 I=1,1SR,1	5078186
TEMP3=(CA1J(11,J)*CQ1J(11,J))/(A3T(J)*A2T(2)*A2T(1)*CH11(11))	5078187
TEMP1=TEMP1+TEMP3	5078188
1861 TEMP2=TEMP2+TEMP3*(1.0-CH11(11))	5078189
XTEMP1=(EPSJIN(J)-A4T(J))/XLAMJ(J)	5078190
XTEMP2=(EBARJ(J)-A4T(J))*TEMP1	5078191
TEMP2=-(GJBAR(J)-A4T(J))*TEMP2	5078192
B(NN,M1)=XTEMP1+XTEMP2+TEMP2	5078193
1880 NN=NN+1	5078194
1890 DO 1940 J=1SCP1,1SS,1	5078195
1900 JJ=11SG-11SF+J	5078196
1910 TEMP1=0.0	5078197
1920 DO 1930 I=1,1SR,1	5078198
1930 TEMP1=TEMP1+CQ1J(11,J)	5078199
1940 B(JJ,M1)=TEMP1/(A2T(2)*A2T(1))	5078200
1950 TEMP1=0.0	5078201
1960 DO 1970 J=1,1SS,1	5078202
1970 TEMP1=TEMP1+A3T(J)*CCPJ(J)	5078203
1980 TEMP1=TEMP1*1SCALE	5078204
1990 B(MX4,MX2)=TEMP1*A1T(3)/A2T(2)	5078205
2000 B(MX4,MX3)=-(TEMP1*A1T(3)*A2T(3))/A2T(2)**2	5078206
2010 B(MX4,MX4)=TSCALE*A2T(1)	5078207
2020 LLL=0	5078208
2030 IF(11SG-11SFP1)2070,2040,2040	5078209
2040 DO 2060 J=11SFP1,11SG,1	5078210
2050 LLL=LLL+1	5078211
2060 B(MX4,LLL)=A3T(J)*(ETAJ(J)-1.0)	5078212

```

2070 XISS= ISS
2080 DO 2100 J=1,ISS,1
2090 KKK=ISSG-IISF+J
2100 B(MX4,KKK)= SHJ(J)-(TEMP1+A1T(3))*2*A2T(3) /A2T(2)
2101 B(MX3,MX4)=A2T(2)/A2T(1)
      B(MX3,MX1)=A2T(2)/A2T(4)
      CALL SSWTCH(3,NSW3)
      IF(NSW3.EQ.1) PRINT 2105,A1T(1),DELY
2105 FORMAT(THOAT Y =E18.8,      5X4MDY =E18.8)
2110 CALL SIMSOL ( B,M,45)
2120 CALL SUB5
2130 RETURN
      END

```

```

5078213
5078214
5078215
5078216
5078217
5078218
5078219
5078220
5078221
5078222
5078223
5078224
5078225

```

SUBROUTINE SUB1	508	1
1000 CALL SSWTCH(1,K000FX)	508	2
GO TO(1001,1005),K000FX	508	3
1001 IF(IRKIND-IDUMP) 1002,1002,1008	508	4
1002 WRITE (2,1003)AIT(1),AIT(2),AIT(3),A2T(1),A2T(2),A2T(3),A2T(4)	508	5
1 ,IRUN,IRKIND	508	6
1003 FORMAT(/////1P7E15.7,3X,A6,I6)	508	7
EOF2=1.0	508	8
1005 WRITE (2,1006)(A3T(J), J=1,ISS,1)	508	9
1006 FORMAT(1P8E15.7)	508	10
IF(IISG-IISFP1) 1008,1007,1007	508	11
1007 WRITE (2,1006)(A4T(J),J=IISFP1,IISG,1)	508	12
1008 RETURN	508	13
END	508	14

SUBROUTINE SUB2	500	1
1000 DO 1001 J=ISFP1,ISG,1	500	2
1001 TAUJP(J)=(TAUAJ(J)/CPP)*CTP**TAUBJ(J)*EXP(TAUCJ(J)/CTP**TAUDJ(J))	500	3
RETURN	500	4
END	500	5

```

SUBROUTINE SUB4
1010 DO 1018 II=1,ISR,1
1011 IF(KFIIND(II))1012,1012,1014
1012 SKFIIN(II)=AKFI(II)*CTP**BKFI(II)*EXP(-CKFI(II)/CTP**DKFI(II))
1013 GO TO 1018
1014 TEMP1=0.0
1015 DO 1016 J=1,ISS,1
1016 TEMP1=TEMP1 +BETA1J(II,J) * SUJ00T(J)
1017 SKFIIN(II)=EXP(-TEMP1-CKFI(II)/CTP**DKFI(II))*AKFI(II)*CTP**
    1BKFI(II)/(CON1*TEMP)**1BETA1(II)
1018 CONTINUE
1019 RETURN
END

```

510	1
510	2
510	3
510	4
510	5
510	6
510	7
510	8
510	9
510	10
510	11
510	12
510	13

SUBROUTINE SUB5	511	1
1000 CALL SSWTCH(1,K000FX)	511	2
GO TO(1001,1018).K000FX	511	3
1001 IF(IRKIND-IDUMP)1002,1002,1018	511	4
1002 WRITE (2,1003)(SUJ00T(J),CCPJ(J),EPSJ(J),SHJ(J),EPSJIN (J),CTVJ(J),	511	5
1,XLAMJ(J),CVJ(J),J=1,ISS+1)	511	6
1003 FORMAT (1P2E15.7)	511	7
1008 WRITE (2,1009)(OF10(II),CKPI(II),CKI(II),SKPIIN(II),	511	8
III),SKBI(II),CH2I(II),II=1,ISR,1)	511	9
1009 FORMAT(1P7E15.7)	511	10
1010 DO 1011 II=1,ISR,1	511	11
1011 WRITE (2,1003)(CQIJ(II,J),J=1,SCPI+ISS,1)	511	12
1017 WRITE (2,1003)(B(J,M1), J=1,M,1)	511	13
1018 RETURN	511	14
END	511	15

```

SUBROUTINE SUB6
  IF(INDSUM).150.1150.1000
1000 DO 1140 J=1,ISS,1
1010 IXXX=MSUMJ(J)
1020 TT1=0.0
1030 TT2=0.0
1040 TT3=0.0
1050 DO 1110 L=1,IXXX,1
1060 XX2=SGJL(J,L)*EXP(-CEJLP(J,L)/AIT(2))
1070 TT2=TT2+XX2
1080 XX1=XX2*CEJLP(J,L)
1090 TT1=TT1+XX1
1100 XX3=XX1*CEJLP(J,L)
1110 TT3=TT3+XX3
1120 EECHJ(J)=TT1/TT2
1130 EECNUJ(J)=ALOG(TT2/SGJL(J,1))
1140 EECCPJ(J)= ( TT2*TT3 -TT1*TT1) /((AIT(2) * TT2) **2 )
1150 RETURN
      END

```

```

S12  1
S12  2
S12  3
S12  4
S12  5
S12  6
S12  7
S12  8
S12  9
S12 10
S12 11
S12 12
S12 13
S12 14
S12 15
S12 16
S12 17
S12 18
S12 19

```

```

SUBROUTINE TEST1
501 IF(IIRKIND-1)1060,1000,1060
1000 IF(IYST)1070,1010,1070
1010 DO 1020 I=1,ISR,1
1020 CHIIT(I)=CHI(I)
      IYST=1
1030 IF(YPREV-AIT(1))1040,1050,1040
1040 CALL WRITE 2
      EXTEST=1.E-7*EXTRAJ(9)
      NOSAY=0
1060 RETURN
1070 IF(AIT(1)-YPREV) 1530,1060,1530
1530 DO 1540 I=1,ISR,1
1540 CHIIT(I)=CHI(I)
1550 DO 1570 J=1,20,1
1560 A4I(J)=A4T(J)
1570 A3I(J)=A3T(J)
1580 DO 1600 J=1,3,1
1590 A2I(J)=A2T(J)
1600 A1I(J)=AIT(J)
1601 A2I(4)=A2T(4)
      IF(EXTRAJ(10).EQ.0.0) GO TO 1620
      IF((EXTRAJ(9)-A1I(1)).GT.EXTEST) GO TO 1621
      EXTRAJ(9)=EXTRAJ(9)+EXTRAJ(10)
      EXTEST=1.E-7*EXTRAJ(9)
1620 CALL WRITE 2
1621 IF(ABS(YSAVE-A1I(1)).LE.EXTEST) GO TO 1622
      IF(NOSAY.NE.0) GO TO 1674
1622 NOSAY=0
1630 IDELXC=IDELXC+1
1640 IF(IDELXC-TEXT1)1670,1650,1650
1650 IF(DELY-EXTRAJ(15))1651,1670,1670
1651 DELY=DELY*EXTRAJ(12)
1660 IDELXC=0
1670 CONTINUE
      IF(EXTRAJ(10).EQ.0.0)GO TO 1676
      IF((A1I(1)+DELY-EXTRAJ(9)).GE.(-EXTEST)) GO TO 1675
1676 CONTINUE
1673 RETURN
1674 DELY=DYSAV
      NOSAY=0
      GO TO 1670
1675 NOSAY=1
      YSAVE=A1I(1)
      DYSAV=DELY
      DELY=EXTRAJ(9)-A1I(1)
      GO TO 1673
END

```

513	1
513	2
513	3
513	4
513	5
513	6
513	7
513	8
513	9
513	10
513	11
513	12
513	13
513	14
513	15
513	16
513	17
513	18
513	19
513	20
513	21
513	22
513	23
513	24
513	25
513	26
513	27
513	28
513	29
513	30
513	31
513	32
513	33
513	34
513	35
513	36
513	37
513	38
513	39
513	40
513	41
513	42
513	43
513	44
513	45
513	46
513	47
513	48

SUBROUTINE TEST2	S14	1
ERR1 = 0.0	S14	2
DO 1020 J=1,ISS+1	S14	3
IF(A3T(J)) 1000,1020,1020	S14	4
1000 ERR1 = 1.0	S14	5
PRINT 1010,A1T(1),DELY,J ,A3T(J)	S14	6
1010 FORMAT(2E15.6,15X,11H GAMMA ,12.9H NEGATIVE,E16.6)	S14	7
1020 CONTINUE	S14	8
IF(IISG- IISFP1) 1061,1030,1030	S14	9
1030 DO 1060 J=IISFP1,IISG+1	S14	10
IF(A4T(J)) 1040,1060,1060	S14	11
1040 ERR1 = 1.0	S14	12
PRINT 1050,A1T(1),DELY,J ,A4T(J)	S14	13
1050 FORMAT (2E15.6,15X,11H EPS ,12.9H NEGATIVE,E16.6)	S14	14
1060 CONTINUE	S14	15
1061 IF(A1T(2)) 1070,1070,1072	S14	16
1070 ERR1 = 1.0	S14	17
PRINT 1071,A1T(1),DELY ,A1T(2)	S14	18
1071 FORMAT (2E15.6,15X,20H T NEG OR ZERO,E16.6)	S14	19
1072 IF(A2T(4)) 1073,1073,1075	S14	20
1073 ERR1=1.0	S14	21
PRINT 1074, A1T(1),DELY ,A2T(4)	S14	22
1074 FORMAT(2E15.6,15X,20H AREA NEG OR ZERO ,E16.6)	S14	23
1075 IF(IRK ID-1) 1091,1080,1091	S14	24
1080 IF (ABS((A1T(2)-A11(2))/A11(2))-EXTRAJ(3)) 1091,1090,1090	S14	25
1090 ERR1 = 1.0	S14	26
PRINT 1081,A1T(1),DELY ,A1T(2)	S14	27
1081 FORMAT(2E15.6,15X,20H T TEST FAILED,E16.6)	S14	28
1091 IF (ERR1) 1170,1170,1100	S14	29
1100 DELY = .3 * DELY	S14	30
IDELXC = 0	S14	31
IFAIL = 1	S14	32
1170 RETURN	S14	33
END	S14	34

```

SUBROUTINE TEST3(F1,F2,I5,I0,N)
  DIMENSION F1(1),F2(1)
  DO 101 I=15,I0
    DEVF=(F1(I)-F2(I))/F1(I)
    IF(ABS(DEVF).GT. 0.1) GO TO 102
101  CONTINUE
    RETURN
102  IFAIL=1
    IDELXC=0
    DELY=.5*DELY
    PRINT 103,A1T(1),DELY,N,I ,DEVF
103  FORMAT(2E15.8,13X,1HA,11,2HT(,12,20H) FAILED MATH TEST -,E16.8)
    RETURN
  END

```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SUBROUTINE OVRFLY
CALL FPT(,TRUE,, 0)
RETURN
END

516	1
516	2
516	3
516	4

```

SUBROUTINE LCOUNT
1000 IF( COUNT-55.0) 1004, 1001, 1001
1001 COUNT=0.0
1002 WRITE (4,1003)IRUN
1003 FORMAT (12H1      RUN :A6)
1004 RETURN
END

```

```

517 1
517 2
517 3
517 4
517 5
517 6
517 7

```

```

SUBROUTINE STOP
1000 IF (A11(1)-YSTOP)1010,1030,1030
1010 IF (EXTRAJ(2)-A11(2))1020,1030,1030
1020 RETURN
1030 IF (ISTART.EQ.0) READ 1,ISTOP
      1 FORMAT(12)
      ISTART=ISTART+1
      PRINT 1063,IRUN
1063 FORMAT(5X,6H RUN ,A6,11H COMPLETED)
1061 CONTINUE
      IF (ISTOP.NE.1) GO TO 160,(5000)
      END FILE 4
1040 IF (EOF3)1060,1060,1050
1050 END FILE 3
1060 CONTINUE
1064 STOP 4
      END

```

510	1
510	2
510	3
510	4
510	5
510	6
510	7
510	8
510	9
510	10
510	11
510	12
510	13
510	14
510	15
510	16
510	17

SUBROUTINE SINSOL(A,KK,LL)	S19	1
DIMENSION A(45, 45)	S19	2
N=KK	S19	3
L=1	S19	4
N1=N+1	S19	5
CALL SSWTCH(3,NSW3)	S19	6
IF(NSW3.NE.1) GO TO 10	S19	7
PRINT 3	S19	8
3 FORMAT(11HOTHE MATRIX)	S19	9
DO 4 IP=1,N	S19	10
4 PRINT 5,(A(IP,JP),JP=1,N1)	S19	11
5 FORMAT(/1X,8E16.8/(17X,7E16.8))	S19	12
10 L1=L+1	S19	13
IF(L-N)21,21,50	S19	14
21 K=0	S19	15
DO 25 I=L,N	S19	16
IF(A(I,L))24,25,24	S19	17
24 K=1	S19	18
GO TO 32	S19	19
25 CONTINUE	S19	20
C DETERMINANT= 0 NO SOLUTION	S19	21
26 PRINT 27, L, IRUN	S19	22
27 FORMAT(5X1A(I,12,18H) = 0, NO SOLUTION,5X4HRUN ,A6.9H SKIPPED)	S19	23
DO 65 IP=1,N	S19	24
65 PRINT 5,(A(IP,JP), JP=1,N1)	S19	25
ISTART=ISTART+1	S19	26
GO TO 160,(5000)	S19	27
32 IF(K-L)26,40,35	S19	28
35 DO 37 J=L,N1	S19	29
B=A(K,J)	S19	30
A(K,J)=A(L,J)	S19	31
A(L,J)=B	S19	32
37 CONTINUE	S19	33
40 DO 41 J=L1,N1	S19	34
41 A(L,J)= A(L,J)/A(L,L)	S19	35
42 A(L,L)= 1.0	S19	36
IF(L-N)43,50,25	S19	37
43 DO 48 I=L1,N	S19	38
IF(A(I,L))44,48,44	S19	39
44 DO 45 J=L1,N1	S19	40
45 A(I,J)= A(I,J)- A(L,J)*A(I,L)	S19	41
46 CONTINUE	S19	42
L=L1	S19	43
GO TO 10	S19	44
50 N2=N-1	S19	45
IF(N2)51,61,51	S19	46
51 DO 60 I2=1,N2	S19	47
I=N-I2	S19	48
I1=I+1	S19	49
DO 59 J=I1,N	S19	50
IF(A(I,J))58,59,58	S19	51
58 A(I,N1)= A(I,N1)-A(I,J)*A(J,N1)	S19	52
59 CONTINUE	S19	53

```

60 CONTINUE
   IF (NSEC.NE.1) GO TO 61
   PRINT 6, (A(IP.NE), IP=1,N)
   FORMAT (SHORES, 7E18.8, / (LX, 7E18.8))
61 RETURN
   END

```

```

0000 00
0000 00
0000 00
0000 07
0000 08
0000 09

```

	FUNCTION DERT(X,Y,I,XD)	S20	1
C	FORWARD GREGORY-NEWTON FORMULA	S20	2
	DIMENSION X(1),Y(1),TX(3),TY(3)	S20	3
	I1=1	S20	4
	I2=I+1	S20	5
	D1=X(I)-X(I1)	S20	6
	D2=X(I2)-X(I)	S20	7
	IF(ABS(D1)-ABS(D2)) 1,2,3	S20	8
2	DO 4 J=1,3	S20	9
	N=I1-1+J	S20	10
	TX(J)=X(N)	S20	11
4	TY(J)=Y(N)	S20	12
	GO TO 7	S20	13
1	DO 5 J=1,2	S20	14
	N=I1-1+J	S20	15
	TX(J)=X(N)	S20	16
5	TY(J)=Y(N)	S20	17
	TX(3)=X(I)+D1	S20	18
	TY(3)=YINT(X,Y,I,TX(3))	S20	19
	GO TO 7	S20	20
3	DO 6 J=2,3	S20	21
	N=I1-1+J	S20	22
	TX(J)=X(N)	S20	23
6	TY(J)=Y(N)	S20	24
	TX(1)=X(I)-D2	S20	25
	TY(1)=YINT(X,Y,I,TX(1))	S20	26
7	FD1=TY(2)-TY(1)	S20	27
	FD2=TY(3)-TY(2)	S20	28
	SD=FD2-FD1	S20	29
	H=TX(2)-TX(1)	S20	30
	R=(XD-TX(1))/H	S20	31
	DERT=(FD1+.5*(2.*R-1.)*SD)/H	S20	32
	RETURN	S20	33
	END	S20	34

C

```

FUNCTION YINT(X,Y,I,XINT)
NEWTON SECOND ORDER DIVIDED DIFFERENCES
DIMENSION X(1),Y(1)
I1=I-1
I2=I+1
D1=(Y(I1)-Y(I))/(X(I1)-X(I))
D2=(Y(I)-Y(I2))/(X(I)-X(I2))
D3=(D1-D2)/(X(I1)-X(I2))
YINT=Y(I)+(XINT-X(I))*D1+(XINT-X(I))*(XINT-X(I))*D3
RETURN
END

```

```

821 1
821 2
821 3
821 4
821 5
821 6
821 7
821 8
821 9
821 10
821 11

```

DIMENSION NUIJP(40,20),NUIJ(40,20),ICAIJ(40,20),EECHJ(20),EECHUJ(20),	S22	1
X0),EECCPJ(20),IBETA(40),IBETIJ(40,20),NUI(40),SPECIK(40,2),	S22	2
X ALPHA(4),BETA(4)	S22	3
DIMENSION ETAJ(20),SBJ(20),THEVJP(20),SHJOP(20),CNJ(20),CXEU(20),	S22	4
XSPECJK(20,4),XNUIJP(40,20),XNUIJ(40,20),ALPIJ(20,20),GJOP(20),	S22	5
XAKFI(40),BKFI(40),CKFI(40),DKFI(40),TAUAJ(20),TAUBJ(20),TAUCJ(20),	S22	6
XTAUDJ(20),EXTRAJ(15),THEVJ(20),SHJO(20),SAJ(20),BETAIJ(40,20),	S22	7
XXNUI(40),BETA(40),A1I(3),A2I(4),A3I(20),A4I(20),AIT(3),A2T(4),	S22	8
XA3T(20),A4T(20),XIA4(20),XIA3(20),XIA2(4),B(45,45),SUJOOT(20),	S22	9
X),CCPJ(20),EPSJ(20),SHJ(20),EPSJIN(20),CTVJ(20),CWJ(20),XLAMJ(20),	S22	10
XTAUJP(20),CVJ(20),C(45,45),CQIJ(40,20),CHI(40),SKBI(40),SKFIIN(40),	S22	11
X),SKFI(40),CKI(40),CKPI(40),DFIO(40)	S22	12
DIMENSION CHIIT(40),MSUMJ(20),SGJL(20,8),CEJLP(20,8),CWI(40),	S22	13
XCZI(40),CAIJ(40,20),KFIIND(40),CEJLPX(20,8),CDI(40)	S22	14
DIMENSION A1IC(3),A2IC(4),A3IC(20),A4IC(20),AITI(3),A2TI(4),	S22	15
I A3TI(20),A4TI(20)	S22	16
COMMON IRUN,ISR,ISS,ISF,ISG,ISC,ICON,INDSUM,JDUMP,DELYP,YSTOPP,	S22	17
ICPOP,RHOOP,CTOP,GAMMA0,CMO,CUOP,CWOP,CLP,EXTRAJ,ETAJ,SBJ,THEVJP,	S22	18
2SHJOP,CNJ,CXEU,TAUAJ,TAUBJ,TAUCJ,TAUDJ,SPECJK,MSUMJ,SGJL,CEJLPX,	S22	19
3GJOP,CHI,CZI,CDI,XNUIJP,XNUIJ,CAIJ,KFIIND,AKFI,BKFI,CKFI,DKFI,	S22	20
4SPECIK,ALPIJ,CCPJ,EECCPJ,EECHJ,EECHUJ,EPSJ,SHJ,SUJOOT,THEVJ,CTVJ,	S22	21
5CVJ,CWJ,EPSJIN,TAUJP,XLAMJ,BETA(4),BETAIJ,CKI,CKPI,CQIJ,DFIO,IBETA(4),	S22	22
6IBETIJ,ICAIJ,NUI,NUIJ,NUIJP,SKBI,SKFI,SKFIIN,XNUI,CHI,CHIIT,	S22	23
7AN,ANFN,BN,SNFN,Z,CORR,CHOB,CHOO,HIND,ALPHA,BETA,CONI,CEJLP,CR0,	S22	24
8SAJ,SHJO,TSCALE,XA,EOF2,EOF3,IEXT11,XB,XC,YSTOP,IISF,IISG,ISCP1,	S22	25
9IISFP1,IISFP1,IISGP1,M,M1,MX,MX1,MX2,MX3,MX4,MX5	S22	26
COMMON COUNT,CPP,CTP,TEMP,XLCT,YPREV,TRATIJ,DELY,IDELXC,IFAIL,	S22	27
1IRKIND,B,C,A1I,A2I,A3I,A4I,AIT,A2T,A3T,A4T,XIA2,XIA3,XIA4	S22	28
COMMON CAOP,IBOP,NOR,I0OP(3),REND(3),REST(3),COP(3,25),IYST,	S22	29
1IGO,ISTART,IPOT,TY(100),TP(100)	S22	30

GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California	(Unclassified)	1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis	GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California	(Unclassified)	1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis
A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.	Unclassified Report		A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.	Unclassified Report	
A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical	I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q.	(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital	A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical	I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q.	(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital
GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California	(Unclassified)	1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis	GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California	(Unclassified)	1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis
A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.	Unclassified Report		A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.	Unclassified Report	
A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical	I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q.	(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital	A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical	I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q.	(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

<p>GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California</p> <p>A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.</p> <p>Unclassified Report</p> <p>A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical</p>	<p>(Unclassified)</p> <ol style="list-style-type: none"> 1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis <ol style="list-style-type: none"> I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q <p>(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital</p>
<p>GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California</p> <p>A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.</p> <p>Unclassified Report</p> <p>A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical</p>	<p>(Unclassified)</p> <ol style="list-style-type: none"> 1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis <ol style="list-style-type: none"> I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q <p>(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital</p>
<p>GM Defense Research Laboratories, General Motors Corp., Santa Barbara, California</p> <p>A FORTRAN COMPUTER CODE FOR INVISCID, NONEQUILIBRIUM STREAMTUBE FLOW by Tung Chen and Alan Q. Eschenroeder. TR65-01P, November 1965. 130 pp., 4 refs.</p> <p>Unclassified Report</p> <p>A Fortran IV computer program for calculating nonequilibrium streamtube flows has been developed. Inviscid gas properties can be obtained for quasi-one-dimensional flows following varying conditions of pressure, density, velocity, or cross-sectional area arbitrarily specified in the streamwise direction. In this program, the kinetic model can include up to 20 species and 40 reversible chemical reactions. The numerical</p>	<p>(Unclassified)</p> <ol style="list-style-type: none"> 1. Fluid dynamics 2. Aerothermodynamics 3. Chemical reactions 4. Fluid flow - Mathematical analysis 5. Hypersonic flow - Mathematical analysis <ol style="list-style-type: none"> I. DA-01-021-AMC-11359(Z) II. TR65-01P III. Chen, Tung IV. Eschenroeder, Alan Q <p>(Descriptors) Chemical reactions, Aerothermodynamics, Fluid flows, Fortran, Digital</p>

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow

integration method has a mathematical accuracy check to determine step size. This eliminates the need for repeated trial runs for optimization. Controlled injection of small perturbations is employed to run the program with some chemical reactions near-equilibrium.

computers, Computers,
Shock waves, Hyper-
sonic flow